CMU/SEI-94-SR-19

AD-A285 047

Draft SEI Program Plans: 1995-1999

August 1994

Special Report
CMU/SEI-94-SR-19
August 1994

Draft SEI Program Plans: 1995-1999

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared for the

SEI Joint Program Office
HQ ESC/ENS
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER

Thomas R. Miller, Lt Col, USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

Table of Contents

List of Figures

Introduction

This document presents the Software Engineering Institute (SEI) strategy and one-year implementation plan for calendar year (CY) 1995, together with the SEI five-year program plan. This document is, in essence, a proposal. It describes the strategic directions and offers detailed options for the coming year. Until the proposed options are selected and budget allocations are approved by the sponsor, the SEI cannot commit to specific work or supporting schedules.

In Chapter 1, we set the strategic context by discussing the SEI charter, mission, vision, strategy, orientation, and customers. The SEI mission is *to provide leadership in advancing the state of the practice of software engineering to improve the quality of systems that depend on software.*

In Chapter 2, we describe the factors that determine SEI plans and set the context for their implementation in support of the SEI mission and strategy. The SEI strategy is to improve software engineering practice by maturing the skills of the software engineering practitioners who develop and maintain software, the managers who organize and lead these activities, and educators who train future generations of practitioners and managers (Maturing the Profession). Our approach to improving the skills of these software engineering professionals is to mature the organizational and managerial processes through which software is acquired, developed, and maintained (Maturing the Process) and to mature the technology used to develop and maintain software (Maturing the Technology). These activities, combined with our core competency in software technology transition, form the strategy for executing the SEI mission.

In Chapter 3, we describe the SEI technical program. We have organized our technical program into clusters of related activities called focus areas to identify and transition those technologies that we believe will most effectively mature the profession, the process, and the technology. The four focus areas are: software process, software risk management, disciplined engineering of software-intensive systems, and trustworthy networks.

1. Through our focus on **software process**, our objective is the maturation of the organizational and managerial processes employed by software development organizations. The SEI seeks to define, model, measure, and improve the maturity of these processes. The expectation is that doing so will improve the organizational performance in developing software.
2. Our technical focus on **risk** provides a systematic and structured process, supported by methods and tools, for identifying, analyzing, and mitigating the uncertainties encountered in a specific software engineering effort. Many of the most serious issues encountered in systems acquisition are the result of risks that remain unrecognized until they have already created serious consequences. The SEI is focusing on risk management because we believe that (a) structured techniques, even quite simple ones, can be effective in identifying and quantifying risk; and (b) techniques exist to mitigate risk.

3. Through our focus on the **disciplined engineering of software-intensive systems**, our objective is the identification and validation of disciplined engineering practices, techniques, and technologies and their transition into software practice. Efforts are focused on two aspects of disciplined engineering: (1) enabling practitioners to share common views and models of architectures for similar systems; and (2) identifying technologies and engineering processes for defining, analyzing, predicting, and controlling performance, reliability, interoperability, and other quality attributes of software systems.
4. Through our focus on **trustworthy networks**, we are concerned with ensuring that computer networks, especially the Internet and eventually the National Information Infrastructure (NII), can be trusted to maintain their own integrity and security and the integrity of the data they transport or store. Drawing on the experiences gained through the Computer Emergency Response Team, the SEI also seeks to mature network security technologies and practices.

Software technology transition is a core competence of the SEI that cuts across and influences all of the activities in the focus areas. The SEI mission requires a technology transition strategy that gives us leverage in meeting the needs of our customers. In Chapter 4 we describe the foundational work that we will do in 1995 that will contribute to the transition of software engineering technology and maintain and enhance this core competence.

Chapter 4 also provides information about the significant roles played by education and services in technology transition. SEI education-related activities aim to improve software engineering practices and advance software engineering as a profession. SEI services help organizations that are influential leaders in the software community to build and sustain continuous improvement of their software processes and their processes for adopting new technologies.

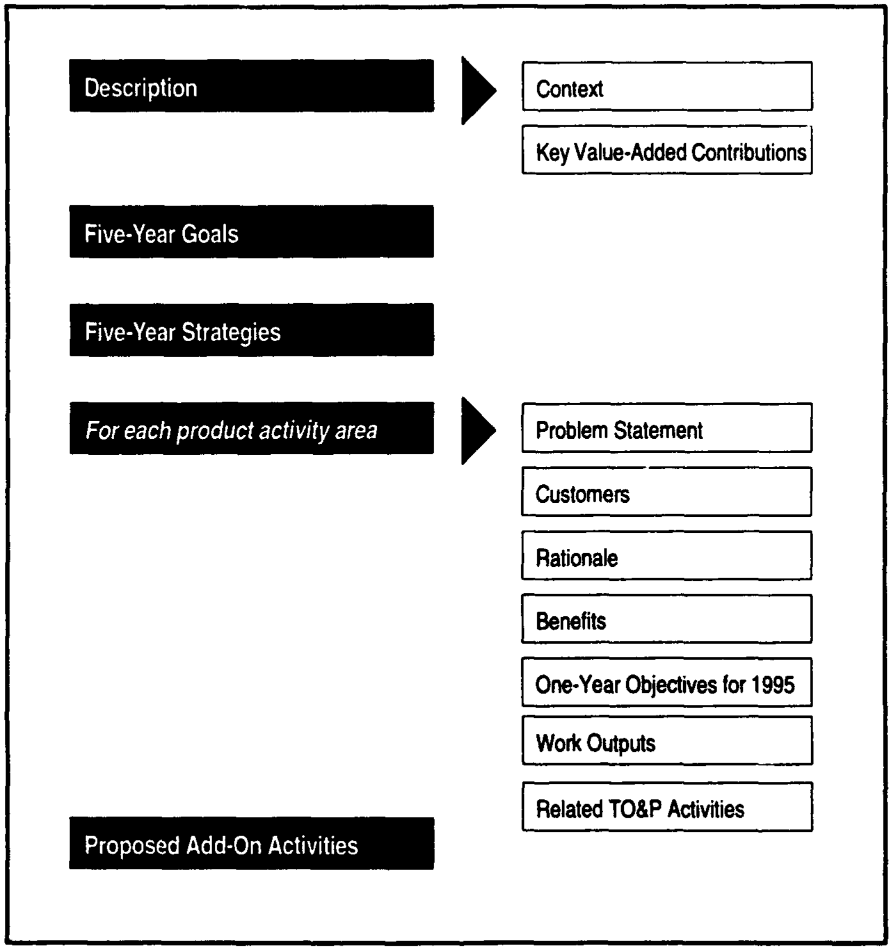
In addition to our own activities, we have a range of other relationships that give our customers opportunities to participate in technology transition activities with us. Chapter 4 provides details about these relationships and about how we involve our customers and keep them informed about our work.

Chapters 3 and 4 discuss the following technical focus areas, areas of expertise, and product activity areas:

Chapter 3: Technical Focus Areas	Focus Areas	Product Activity Areas	Page
	3.2 Software Process	3.2.4 Process Maturity Modeling	42
		3.2.5 Process Definition	48
		3.2.6 Process Measurement	53
	3.3 Risk Management	3.3.4 Risk Management in Systems Acquisition and Development	72
	3.4 Disciplined Engineering	3.4.4 Software Architectures and Models	97
		3.4.5 Product Quality Engineering	103
		3.4.6 Automation of Engineering Practice	108
		3.4.7 Maturation of Engineering Practice	115
	3.5 Trustworthy Networks	3.5.4 Incident Handling	134
		3.5.5 Incident Prevention	136

Chapter 4: Technology Transition	Areas of Expertise	Product Activity Areas	Page
	4.2 Education in Technology Transition	4.2.4 Educational Product Development and Delivery	161
		4.2.5 Educational Infrastructure	169
	4.3 Services in Technology Transition	4.3.4 Software Engineering Improvement	188

The sections on the four SEI focus areas in Chapter 3 and the two areas of expertise in Chapter 4 each follow a common structure:



This year the SEI is seeking customer evaluations of proposed new 1995 work during this early stage of program formulation. Accordingly, this 1&5 Year Plan contains two categories of 1995 core project work:

1. Baseline Core Projects: These are core projects that represent multi-year work in progress that we are committed to implement in 1995, and work that is necessary to complement ongoing commitments in related technical objectives and plans (TO&P) programs. These activities are described generally within the focus area sections entitled "One-Year Objectives for 1995" and "Work Outputs."
2. Add-On Core Projects: These are proposed new core work activities for 1995, and generally represent discretionary new starts and directions for the 1995 core program. The SEI is asking that selected external evaluators provide their inputs on these new technical directions by evaluating the merits of all proposed core add-on projects. The add-on activities are contained in sections entitled "Proposed Add-On Activities" at the end of each focus area description.

Based on expected 1995 core funding levels, approximately half of the proposed add-on projects can be supported by the core program in 1995. Thus, external customer inputs can play an important role in shaping priorities within the core program. Appendix A lists all baseline and add-on items.

Chapter 1: Table of Contents

1 Strategic Context

The purpose of this chapter is to describe the strategic context for the five-year plan and one-year implementation.

The Software Engineering Institute (SEI) was established in 1984 by Congress as a federally funded research and development center with a broad charter to address the transition of software engineering technology. The SEI is funded by the Advanced Research Projects Agency (ARPA) through a contract with the Air Force Materiel Command/Electronic Systems Center, and through additional contracts with other sponsors, clients, and partners. These relationships determine organizational, funding, and reporting structures as well as providing a comparative advantage and natural focus for selecting customers and activities.

As an integral component of Carnegie Mellon University (CMU), the SEI maintains a highly qualified staff and conducts its activities in a manner commensurate with that of the university. As a member of the CMU community and as an ARPA-funded organization, the SEI is an active participant in the software research community at large.

1.1 Charter

The SEI charter is to:

- Provide the means and leadership to bring the ablest professional minds and the most effective technology to bear on rapid improvement of the quality of operational software in software in software-intensive systems.
- Accelerate the reduction to practice of modern software engineering technologies.
- Promulgate the use of this technology throughout the software community.
- Foster standards of excellence for improving software engineering practice.

The SEI is funded by a combination of core from ARPA and direct support from specific government customers. The core enables the SEI to engage in a combination of research, education, technology exploration, and development of transition products and services to achieve broad technology transition. The SEI may receive funding from federal agencies other than ARPA for specified work consistent with the charter.

1.2 Mission, Vision, and Strategy

Software represents an enormous opportunity for cost-effective flexibility in military and commercial systems. Historically, our customers have experienced significant difficulties in acquiring, deploying, and maintaining large-scale software systems. Acquired software often does not meet expectations, is delivered late and over budget, and is difficult to change to meet evolving needs. We believe that these problems can be avoided by bringing an engineering discipline to the way software is created. However, the current state of the practice is far behind the state of the art. Technology transition is the means of closing this gap.

Our **mission** is to provide leadership in advancing the state of the practice of software engineering to improve the quality of systems that depend on software.

We want our customers to be capable of applying a mature software engineering discipline to produce high quality software that meets their expectations, at a competitive price and on predictable schedules. Therefore, we are committed to the evolution of software engineering from an ad-hoc, labor-intensive activity to a managed, technology-supported engineering discipline.

We **envision** ourselves as an organization dedicated to identifying and opening gateways to improved software engineering practice. We see ourselves in the role of enablers, improving the practice by establishing human and technology connections that will reduce obstacles to technology transition and allow improved practices to spread throughout the industry.

Our intent is to identify and transition to customers, through transition products and services, those processes, methods, and tools that will help them make lasting improvements to their overall software engineering capabilities.

Our **strategy** for implementing this intent is to improve the state of the practice of software engineering by maturing the software engineering profession (Maturing the Profession). This strategy is based on maturing the skills of the software engineering practitioners who develop and maintain software, the managers who organize and lead these activities, and educators who train future generations of practitioners and managers. Our approach to improving the skills of these software engineering professionals is to mature the organizational and managerial processes through which software is acquired, developed, and maintained (Maturing the Process) and to mature the technology used to develop and maintain software (Maturing the Technology). These activities, unified by our core competency in software technology transition, form the strategy for executing the SEI mission. (See Section 2.3 for more information on the strategy.)

In applying this strategy, we will focus our technical activities in software engineering technology areas of critical importance to our customers. We will continue to address other important software engineering issues, but will not seek to establish a leadership position in those other areas. To accomplish a leadership position in an area of focus requires at least 25-30 people with appropriate expertise, and an additional cadre of specialized support. With our size constraints, we cannot expect to focus in more than five areas. A minimum of four areas appears necessary to have the broad impact prescribed in the charter.

In each focus area, we research, evaluate, mature, and demonstrate technology solutions in a realistic environment. Demonstrations are planned so that (1) the SEI can determine whether a product or service should be developed, (2) risk of adoption is reduced in the eyes of potential customers, and (3) the costs and benefits of adoption are measured to support an acceptable return on investment to SEI customers.

Also in each focus area, we identify (1) the customer base; (2) customer strategic intent, needs, and requirements; (3) our vision, goals, and objectives; and (4) the specific products we will develop to achieve those goals and objectives.

Our products and services include courses, events, publications, prototype software, videotapes, and guidance and advice in the use of our products. These products and services are intended to help the software community improve its management practices, technical practices, and the capabilities of its personnel.

1.3 Orientation

As a technology organization, the SEI promotes software engineering and supporting technology. Technology is our strength, and we must be technology driven. However, we do not promote technology for its own sake; we are also needs driven. A need can result from an encountered problem, an opportunity enabled by innovation, or anticipation of future problems or technological advances. We help organizations understand the root causes of their software engineering problems as needs. Four considerations influence which problems we work on:

1. The mission to advance the state of the practice of software engineering requires the SEI to have a broad impact by concentrating on those problems that are pervasive.
2. The SEI is in a trusted position that demands objectivity. Organizations expect the SEI to exert independent technical judgment and influence based on a broad and deep understanding of the field, and to understand and provide solutions to the root causes of problems, not simply to eliminate a symptom.
3. The SEI is a relatively small organization. More needs and problems exist than we can address, and there is more work to be done than we can expect to accomplish. We must be selective in choosing problems that are strategically important and have high-leverage potential, understand where outside expertise is available, and work within our abilities.
4. The SEI, by contract, is not permitted to compete in markets predictably and properly satisfied by commercial enterprise.

We are committed to be a needs-driven organization in this sense and have made this orientation an explicit part of our business. We will pursue technologies that offer solutions to real needs. To have a broad impact, we will provide solutions in the form of products and services that help organizations help themselves.

1.4 Customers

Customers are beneficiaries of SEI products and services. The SEI has customers in the Department of Defense (DoD), in other federal agencies, and in industry and academia. The latter develop much of the DoD software and train software practitioners. To provide better service, we have identified three special categories of customers **(sponsors, clients,** and **partners)** with whom we collaborate in the development, maturation, and initial transition of needed products and services.

1.4.1 Sponsors, Clients, and Partners

Figure 1-1 shows some of the SEI interactions that help us perform our mission of improving software practice.

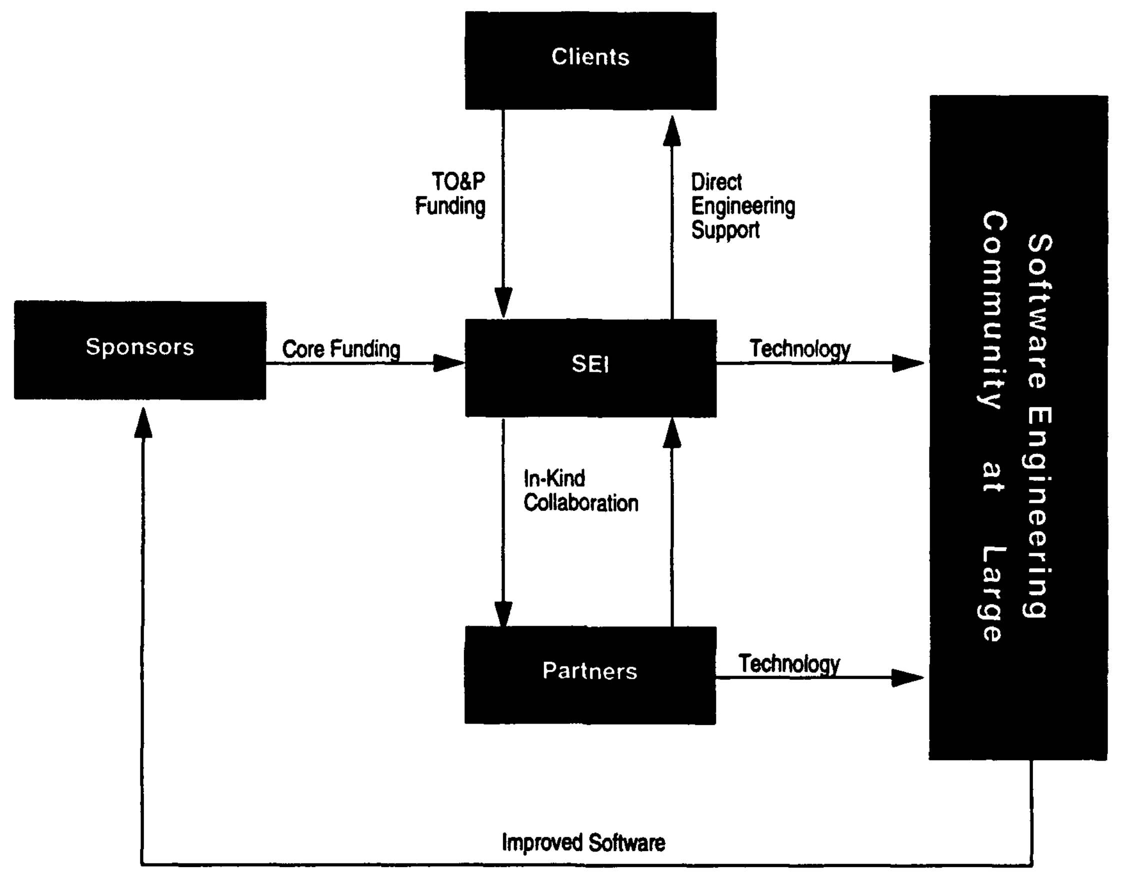


Figure 1-1: Interaction with Sponsors, Clients, and Partners

Sponsors invest funds in the development of capability. For instance, a sponsor might fund the SEI to investigate a certain technology. Sponsorship may be tied to the condition that the sponsor be the initial customer of a resulting product. A specific SEI activity could have multiple sponsors. We work with our sponsors to ensure that we are working on the right problems and to get their support for our approach and plan.

The DoD, through ARPA, is our major sponsor and invests funds in the SEI base (core funding). This enables the SEI to understand needs, evaluate technology, propose and test solutions, and then to develop and demonstrate products and services for our customers. These base funds also enable the SEI to develop and maintain relationships with the supporting software infrastructure in the United States.

Clients work with the SEI to address specific software engineering problems. A significant portion of the SEI's total resources is received through technical objectives and plans (TO&P) funding from clients. Whereas core funding enables the institute to investigate emerging ideas and technologies, TO&P agreements provide the means for the SEI to proof-test or transition promising results into practice for specific customers. This type of interaction establishes a near-term conduit for SEI products and services to flow into the software community, and it permits the SEI to maintain insight into the nature of software practice. Through TO&P agreements, the SEI works in the field to promote and verify improved practices in conjunction with the sponsor and to gather data that will influence future efforts.

Partners invest resources, including funds and people, to collaborate in the development, demonstration, or transition of SEI products or services. They may benefit directly or indirectly by the partnership. They also assume some risk. They contribute to the success of a specific product by providing expertise, perspective, credibility, and/or delivery capability. Organizations that send resident affiliates (that is, individuals on long-term assignment at the SEI from their home institutions) are, by definition, partners.

Partners provide us with insight into problems, assist with testing SEI products, or offer a context for demonstrating solutions. The primary consideration in matching our capabilities to specific partners' needs is the credibility partners bring to the test or demonstration. They should bring specialized expertise to the SEI or be representative of a class of potential customers. They will be selected based on their contribution to the success of an activity, their relative importance to an SEI sponsor, or their contribution to the sponsor. In addition, commercial vendors may provide leverage for SEI products by becoming transition partners who service broader markets than the SEI would be able to serve.

1.4.2 Acquisition, Development, and Post Deployment

Our DoD customers focus on three distinct phases of the software life cycle: (1) acquisition, (2) development, and (3) post-deployment support. Each phase generates somewhat different software engineering concerns.

Acquisition is the phase in which requirements are defined and contracts are let for software development to meet these requirements. Concerns of acquisition organizations include policy, standards, requirements definition, cost and schedule estimation, contract and risk management, reengineering, reuse, training, and testing.

Development is the phase in which software is created that satisfies the requirements of the contracts resulting from the acquisition phase. The concerns of development organizations include requirements, specification, design, coding, integration, testing, risk management, installation, training, and project management.

Post-deployment is the phase that addresses the support of the software after the system is fielded (operational). The principal concerns of post-deployment software support (PDSS) organizations are reliability, maintainability, reengineering, and the costs associated with these.

1.4.3 Managers, Practitioners, and Educators

In serving our customers, we have identified that the needs of managers, practitioners, and educators differ, and we have tailored our product offerings accordingly. In the following section, we describe our understanding of those needs from the perspective of our principal customers.

Managers concentrate on system acquisition that encompasses all phases of the life cycle: research, development, production, and operation. Acquisition is performed by an organization representing the end user of a software-intensive system.

Each military service has program executive officers (PEO) who serve as system materiel developers responsible for acquisition of the system. PEO organizations are co-located with supporting functional commands and have small staffs. Mission accomplishment is through the use of the matrix concept, where functional services and expertise are supplied by supporting functional commands, e.g., life-cycle software engineering centers. System development is generally performed for the PEO by industry through a contract. The PEO is responsible for the management of the system development through a statement of work that is specified within the contract.

Generally, the responsibility for the systems software technical management of a life-cycle software support center is assigned to a program manager (PM). Contractor management is responsible to the PM for complying with the specifications within the contract and responsible for managing the technical development of the system in the most cost-effective way while ensuring high-quality software.

The industrial counterpart to the military PEO is the senior executive (typically division or site manager) responsible for the development of the contracted system. The SEI promotes the acceptance and use of methodologies that bring a higher degree of management control to the contractor's development processes while reducing the degree of technical risk at crucial points in the development cycle.

The guiding principle for the SEI is the belief that acquisition and development should function more as a partnership than as a traditional business venture because the resulting system frequently involves life-critical functions supported by technology that is important to the nation as a whole; it is not a simple profit-seeking activity for the benefit of a single business enterprise. Therefore, the SEI promotes the best practices for effective management by both parties to the contract.

Practitioners are responsible for both pre-deployment and post-deployment total life-cycle software support. Government software engineering centers provide technical support to the PEO throughout the acquisition phase and the end item manager for the remainder of the system life cycle. These centers assist the PEO in ensuring that the software being developed for

the system can be supported by internal resources or contractor support. Once the system is operational, the center is then responsible for software development support through enhancements and refinements that generally result in software version changes on a cyclic basis.

The SEI brings its effort to bear on identifying, evaluating, and disseminating methods, tools, and techniques that suggest a significant improvement over traditional approaches to system development. The intent is to bring to the acquisition team a greater ability to articulate software system requirements and to have developers equipped with the best knowledge and technologies currently available. This dual support for the acquiring government agency and the industrial contractor represents the best way to affect the overall set of life-cycle concerns.

Support for software development activities includes providing the practitioner with methods that ensure consistently high quality results in terms of system performance. Such methods address issues ranging from requirements analysis through design, coding, and test and integration. Further support to the practitioner comes in the form of tools that help automate certain aspects of particular methodologies. The goal is to provide the practitioner with an integrated set of methods and tools that will enable consistent results for the individual, the project team, and the suite of projects within the parent organization.

Post-deployment support generally corrects latent defects and performs enhancements to add greater functionally for incorporating new requirements to the existing system. The support centers are augmented with support contractors that will provide software development for each system that is in post-deployment. This contractor service augments the government practitioners who are responsible for the development of the version and block changes to each system. Practitioners that work within the PDSS community must benefit from the application of disciplined software engineering and the creative use of existing technology. Reengineering will be a new process that will help to improve productivity within PDSS.

Educators (and trainers) are responsible for meeting the nation's need for well-qualified software engineering professionals. In addition, education and training are essential components of technology transition.

By charter, software engineering education is part of the SEI mission. To better prepare new and existing software engineers to perform high-quality software development, the SEI must accelerate the development of software engineering programs in academic institutions. However, the education need is not limited to the academic sector. To improve the capability of current software practitioners, the SEI must enhance the quality and availability of continuing education and training programs in government and industry. Our efforts will be successful to the extent that the education infrastructure prepares individuals to participate in the software engineering activities of our customers in industry and government.

Chapter 2: Table of Contents

2 Strategic Overview

Software has become critically important to both our national defense and economic survival. It pervades our entire society, providing more and more of the functionality previously provided by hardware and expanding the capacity of hardware for multiple applications. As a result, the strategic importance of the Software Engineering Institute (SEI) mission to provide leadership in advancing the state of the practice of software engineering cannot be understated.

Our approach to improving software engineering practice is set in the strategic framework of maturing the software engineering profession. By this we mean maturing the skills of the practitioners who develop and maintain software, the managers who organize and lead these activities, and the educators who teach future generations. This is accomplished by maturing the organizational and managerial processes and the technology to develop and maintain software. The strategic framework is unified by our core competency in the area of software technology transition.

This chapter describes the strategic factors that determine SEI plans and sets the context for their implementation in support of the SEI mission. Section 2.1, Situation Analysis, provides an analysis of the current political and economic situation, and describes the major trends that are projected to significantly impact the field of software engineering and the SEI over the next five years. Section 2.3, Strategy for Improving Software Engineering Practice, describes the strategic framework which unifies the SEI's activities in support of its mission over the next five years, its selected core competency, and the rationale for its selection. Section 2.4, Planning Considerations, specifies constraints considered by the SEI in defining its technical program. Section 2.5, Conclusion, summarizes the strategic context that forms the basis for the SEI technical program described in Chapter 3.

This plan continues to be based on the assumption of a continuing Department of Defense (DoD) "no-growth" strategy for the SEI during the five-year planning period. It reflects a commitment to effective cost control, increased leverage of resources, and focused efforts in those areas that will provide the highest payoff to SEI customers. At the same time, the plan reflects the support of the SEI for evolving national priorities resulting from the changing world political and economic environment.

2.1 Situation Analysis

The U.S. has undergone a paradigm shift as a result of dramatic geopolitical events of the early 1990s, such as the dissolution of the Soviet Union and the Clinton Administration's national focus on strengthening the economy. As a result several trends have emerged that will affect the SEI technical plans significantly during the five-year planning period.

Third-world countries that are increasing their military strength and boldness, the changing and undefined regionalized military threat, and the renewed focus on national competition at the global level are influencing the Clinton Administration's national priorities and emerging

policies. The U.S. economy has undergone a complex transformation; and as a result, there is a new linkage between technology and economic policy. This transformation is caused, in part, by the shrinking defense industry; the transition to just-in-time inventory and total quality practices; and the continuing belt-tightening and adaptation of global competition. The DoD can no longer afford its own separate technology base. There is an emerging equivalence of national defense security and economic security, and the priority to create a strong integrated technology base to support both. To implement these policies, the administration has developed the National Technology Policy (NTP); changed the role of the Advanced Projects Research Agency (ARPA); initiated the Defense Conversion Program, the Technology Reinvestment Project, the National Information Infrastructure (NII), and the Advanced Technology Program (ATP) through the National Institute for Standards and Technology (NIST).

The Clinton Administration's technology initiative has three basic goals. The first goal reflects the objective of economic growth to both create jobs and protect the environment. The second goal is to make the government more efficient and responsive; and the third goal is to re-affirm that the U.S. will maintain its world leadership in basic science, engineering, and mathematics, at the bench and in the classroom.

In the face of the new world order, the administration is improving U.S. economic strength through policies that invest more funds in the national industrial base, revitalize the national infrastructure, improve education in math and science, protect the environment, shift defense-based research and development (R&D) programs to the industrial sector, and achieve global competition.

These policies are illustrated by the reduction of DoD budgets and the downsizing and changing role of the military, the changing roles of defense and civilian resources, the increasing concern for the domestic infrastructure, and the increase in global competition. Clearly, support for these goals will significantly influence the long-term direction and technical focus of the SEI.

2.1.1 DoD Budget Reductions, Downsizing, and the Changing Role of the Military

Because of budget cuts, the DoD's claim on the U.S. technology base will continue to decline. National defense capability will become more and more dependent on technology that is first developed and applied in the commercial sphere. Budget and armed forces reductions have caused a decrease in DoD organic capabilities and contractor bases, a growing need for increased flexibility, a concern for system evolution, and a need for acquiring systems more efficiently. Much of this will result in pressure to maintain existing systems and components for longer periods, creating more dependence on reuse and reengineering for extensive and responsive system modifications. Fewer new systems will be built, and existing systems will need to be evolved to meet new threats. Simulation will become an even more cost effective method for military training and system evaluation. Commercial off-the-shelf (COTS) products will play a larger role in rapid, flexible system integration for supporting military readiness in

regional and global conflicts. The integration of software COTS products into systems to support military needs must become a strength of the industry supporting defense.

In the aftermath of the Soviet Union, many third-world countries have increased their military strength. More of these nations will be able to develop or acquire nuclear potential, medium-range ballistic missile delivery systems, and chemical or biological weapons. The political instability in most of these countries gives cause for alarm, in addition to, extra-national groups (e.g., those involved in illegal drug trafficking) that are well funded and equipped and willing to engage in military and political activities. The undefined nature of this regional threat requires the rapid development and deployment of systems that support the changing mission of the military from rapid responses to both military threats and humanitarian needs. Furthermore, a new paradigm for logistic support and command and control will be needed. Recent domestic and international natural disasters have reemphasized the role the military plays in humanitarian relief and assistance. This role requires improvement in humanitarian logistic support and in command and control for domestic emergency management.

2.1.2 The Changing Emphasis of Defense and Civilian Resources

With the passing of the Cold War, it is clear that a portion of the nation's resources, formerly devoted to defense, must be shifted into other more economically productive areas. Defense conversion has been defined as the process by which people skills, technology, equipment, and facilities in defense are shifted into alternative economic channels. It is not just simply a way of saving industries with a declining defense base, it is an important national strategy to make more productive use of the nation's resources. Currently, a move towards increased convergence between commercial capabilities and defense needs is underway. ARPA's Technology Reinvestment Project is a collaborative effort of five different agencies that could be a model for future interactive programs among agencies. The Department of Commerce has also put increased emphasis on technology and has created two programs within the NIST: the ATP and the Manufacturing Extension Partnership Program. It has also increased NIST's budget over the coming years.

Accompanying these changes is an increased awareness of the need for acquisition reform. Today, companies are often not willing to integrate commercial and defense practices because they cannot afford the additional overhead costs created by current DoD acquisition policies. ARPA recognizes that the only way to effectively achieve some of their goals in technology reinvestment is to change some of these acquisition policies.

2.1.3 The Increasing Concern for the Domestic Infrastructure

The NTP was developed to rejuvenate the nation's economic infrastructure and to revitalize the national industrial base. This policy will focus on increasing R&D and technology transition in the commercial sector; development of information technology as an engine of economic growth; and improvements in manufacturing technology, health care, transportation, communications, and education. ARPA and the DoD, through NIST, will play an increasingly impor-

tant role in supporting the policy through the development and deployment of dual use technologies.

The increasing importance of software engineering in support of the NTP's focus on technology transition for improved national competitiveness will most likely shift the national R&D focus toward improving manufacturing technologies, upgrading the national transportation system, implementing the NII and enhancing commercial communications, and addressing software reliability in critical medical applications for patient monitoring, treatment, and medical imaging.

2.1.4 The Increase in Global Competition

International competition will intensify as world industrial and technological capability is distributed among industrialized nations. Internationalization of economic and technological activity will deepen the interdependence between national economies and lessen the line between domestic and foreign policies. The focus on international competition will reinforce the need for international standards and highlight the increase of sophistication in global technology. Hence, the SEI will be required to increase its international participation with the technical community; assist in the development of the standards required for doing business in this environment; interpret the impact of these standards on our own economic base; and advocate the formulation of a responsive national policy for effective competition in the global marketplace.

2.2 Effect of Major Trends on Software

The federal budgeting process has begun a shift that reflects the changing national priorities described previously. The boundary between the mission of the military sector and the relevance of the work to civil sector opportunities will largely disappear. Federally funded laboratories will spend more time on developing an understanding of the relevance of their work to the needs and opportunities of non-DoD agencies and civil sectors.

This will be accompanied by a shift from defense-related R&D to civil sector R&D. In terms of the ratio of investment between civilian and military R&D, the intermediate target is a ratio of about 50/50. As a result, the SEI expects to see no more than the same level of funding support from the DoD. At the same time, we can expect increased need and funding support from other federal agencies as they struggle to overcome the same software-related problems that have been addressed effectively within the DoD over the past decade.

Government agencies will need to acquire, develop, and maintain software-intensive systems more efficiently as the number of systems to maintain increases and the newly acquired systems become more complex. In combination with declining budgets, this need will require improvements in the process by which organizations buy software-intensive systems. The DoD's current system acquisition model stems from a hardware orientation. It assumes that requirements are known up front, and that systems can be built exactly as they are initially conceived.

A measurable process that supports successful management of software-intensive systems acquisition also needs to be developed.

Use of products and standards emanating from the commercial world offers an attractive way for the DoD to acquire and evolve systems of high quality at minimum cost and risk. The present DoD acquisition system is not flexible enough to adequately incorporate COTS and advanced technology demonstrations, nor does it have the means to reduce schedules sufficiently to field a system that uses current state of the technology. DoD system acquisition procedures may change to allow more frequent use of COTS products. In using COTS, industry standards will become even more important to the DoD. Hence, the SEI must focus on the capability to design systems and their architectures using these standards.

Increased national competition will require shorter system development times to meet unforeseen requirements. This suggests that future systems will be created and configured on demand from proven concepts, architectures, and components. This situation will place greater emphasis on software architecture, reuse, and reengineering in the short term and design for reengineering and automatic program generation in the long term. It will also require more effective software engineering practices that can be applied much earlier in the system life cycle than current practices.

Defense planners will have to create smaller and better-trained forces, supported by high-performance equipment that is adaptable to changing threats. The equipment and systems that support better training are increasingly dependent on software for their functionality. Concurrently, computing power, because of advanced semiconductor technology, continues to double about every four years. These trends create demands for affordable, reliable, and flexible software that are more and more difficult to satisfy.

As the DoD downsizes, the emphasis on improving the U.S. economy and infrastructure moves the nation toward a more commercially oriented R&D base, with focus on technology transition to the commercial sector. The result of this is that increasing amounts of research relevant to the DoD will be conducted by other federal agencies or industry. Hence, the SEI must provide more support to these other federal agencies and industry to participate in this broadened range of dual use technology developments relevant to DoD software engineering.

The reduction in operational funding for the military will also emphasize the importance and cost effectiveness of computer simulation for training and system evaluation. The increased use of real-time simulation, both for defining and refining system requirements and for training, suggests that there will be a need for software that can meet time constraints within a network environment. It also suggests the need for vastly improved human interface technologies to provide the required realism for effective evaluation and training.

The need to respond quickly in a worldwide theater of operations suggests increased portability of command-control and intelligence facilities. It also suggests a greater use of telecommunication technology, such as teleconferencing and telepresence, so that people with much

needed skills who are located remotely can be put to use solving local battlefield or humanitarian relief problems.

Budgetary constraints and rapid changes in military strategy are also creating the need for more flexible manufacturing capability. DoD acquisition may fund prototype development and defer full-scale production until its need is clearly demonstrated. The dual use nature of this emphasis on "agile manufacturing" for both defense and commercial needs implies that the SEI should devote more attention to processes and tools for supporting manufacturing technology transition efforts.

The SEI must respond to the changing political and economic environment and the need to revitalize the national infrastructure, particularly the nation's industrial base. We should increase our focus on efficient acquisition, development, and maintenance of software-intensive systems, taking into consideration dual-use software technology. We should support the development of enhanced software architectures; improved techniques for reuse and reengineering; real-time simulation; expanded and more flexible communications capabilities; system integration of commercial software; and software engineering processes and tools for advanced manufacturing technologies. To develop the underlying foundation for our technical focus areas, we should continue our work in education and training and improving the state of the practice of software engineering through improved processes and software risk management.

2.3 Strategy for Improving Software Engineering Practice

The current political and economic situation, as described in Section 2.1, clearly establishes the importance of software to the defense and economic well being of the nation. Because software pervades nearly every aspect of society, it is vital to address effectively the issue of continuous improvement of the practice of software engineering as an essential ingredient of our national strategy. With this in mind, it is important to articulate clearly the strategic framework which supports the SEI mission to improve the state of the practice of software engineering.

Figure 2-1 summarizes the SEI strategic framework to improve software engineering practice in support of our national strategy. It is through this strategic framework that the mission of the SEI will be executed effectively.

The SEI strategy for improving software engineering practice is to mature the skills of the soft ware engineering practitioners who develop and maintain software, the managers who organize and lead these activities, and educators who train future generations of practitioners and managers (Maturing the Profession). Our approach to improving the skills of these software engineering professionals is to mature the organizational and managerial processes through which software is acquired, developed, and maintained (Maturing the Process) and to mature the technology used to develop and maintain software (Maturing the Technology). These activities, combined with our core competency in software technology transition, form the strat-

egy for executing the SEI mission. Software technology transition has been chosen as our core competency because of its centrality to the SEI mission.

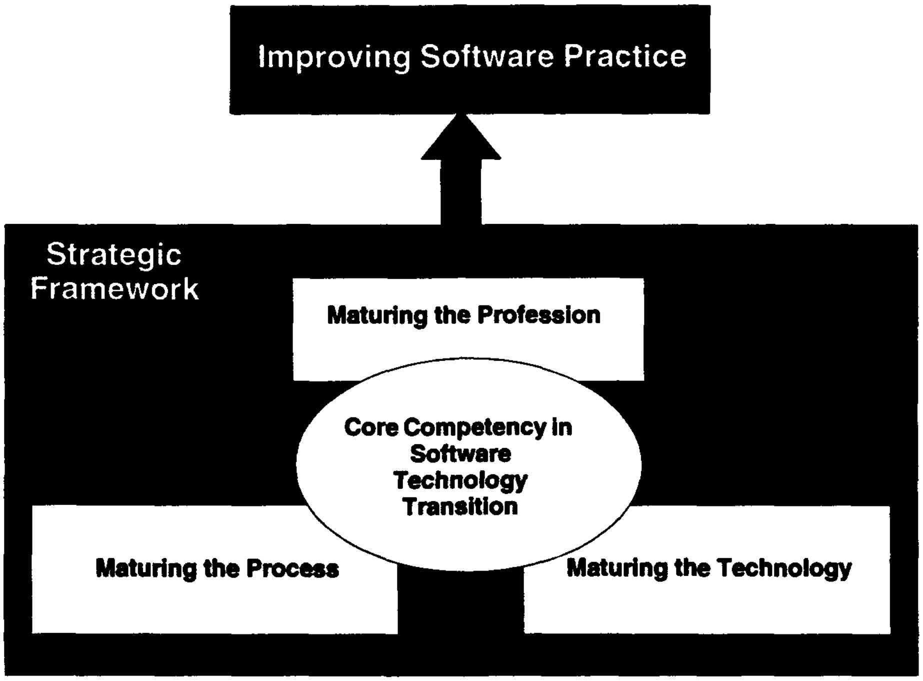


Figure 2-1: Strategic Framework for Improving Software Practice

2.3.1 Maturing the Profession

Software practice is performed by people using processes, methods, and tools that they and their predecessors have created. These people compose the software profession, and it is through them that the SEI intends to improve software practice. To do this, the SEI aims to mature the skills of software engineering practitioners, managers, and educators. Our approach to improving the skills of these software engineering professionals is to mature the processes through which they develop and maintain software, the technology they use to develop and maintain software, and the educational infrastructure that supplies the practitioners and managers. Improving the skills of these software professionals will result in improved effectiveness and efficiency in the state of the practice of software engineering. For details on how the SEI transitions these improvements into the profession, see Chapter 4.

2.3.2 Maturing the Process

The Capability Maturity Model (CMM) was conceived as a model for judging the maturity of the organizational and managerial processes of an organization and for identifying the key practices that are required for maturing these processes. Through the CMM, the SEI has put in place an effective means for modeling, defining, and measuring the maturity of the organizational and managerial processes used by software professionals. CMM-derived instruments like prSoftware Process Assessments (SPAs) and Software Capability Evaluations (SCEs) allow a software development organization's process maturity to be appraised. With SPA or SCE results acting as benchmarks, organizations can then use key process areas (KPAs) and best practices to improve the maturity of these processes. In 1994 the SEI unified SPA and SCE through the CMM-Based Common Rating Framework, which ensures that SPA (now called Internal Process Improvement) and SCE results will be consistent with each other and with the CMM.

2.3.3 Maturing the Technology

More mature organizational and managerial processes are necessary but not sufficient for a mature software engineering profession. SEI experience suggests that where organizational and managerial processes are at CMM level 3 or higher, the need for more mature technology becomes evident. Particularly needed are technologies that allow quality attributes such as performance, reliability, timeliness, dependability, and trustworthiness to be reliably and accurately predicted and controlled. Also needed are commonly accepted system and subsystem architectures and models that operate at different levels of abstraction.

To address the needs for more mature technology, the SEI in 1994 conducted a study to determine the feasibility of an Engineering Maturity Model (EMM) to address the maturing of technology and engineering practices in a manner analogous to the CMM. The results of the feasibility study suggest that this is possible, and development of the EMM will commence in 1995. The EMM development is described in more detail in Section 3.4.

2.3.4 Core Competence in Software Technology Transition

By software technology transition, we mean movement of the best software engineering processes, methods, and tools from R&D into broad use in the software engineering community. The SEI is a value-adding transition agent between researchers whose results can improve software practice and practitioners who can apply this result to solve important and pervasive software problems. The SEI adds value by identifying relevant research results and making them understandable and applicable by practitioners, and by identifying root causes of problems faced by practitioners and making them understandable and applicable to researchers. Thus, while the computing research community aims to advance the state of the art, the SEI aims to incorporate state-of-the-art advances into the state of the practice. Through our interactions with practitioners and researchers, the SEI seeks to identify "best practices" and to promote widely their introduction into the practice of software engineering. Successful technology transition results in overall improvement in the state of software engineering practice.

To build its core competency in transition, the SEI adapts transition models from other disciplines and applies them to software. These models help us approach technology transition in a systematic and effective way. We identify transition methods and transition vehicles that facilitate adopting and institutionalizing improved processes, methods, and tools. We develop transition products and services that help people help themselves improve their practices.

Because the size of the SEI technical staff is limited, we seek leverage for our transition efforts. We gain leverage by influencing software engineering education and providing educational materials that aid the teaching of good software engineering practices, by providing services—primarily advice and guidance to government organizations—that aid continuous improvement efforts, and by working with transition partners who can take our products and services to the community at large.

2.4 Planning Considerations

In developing plans within the context of the changing political and economic environment, the SEI must consider several factors. The most significant of these planning considerations are:

- The mission requires that the SEI facilitate the transition of appropriate software technology into practice for the mutual benefit of the DoD and other federal agencies in support of national priorities and objectives.
- The SEI strength is in the area of software technology—broadly defined to include traditional "computer science" and the evolution and transition of engineering practice. The SEI must maintain its focus on technology to provide the stability that is vital to an R&D program, and must emphasize efforts that result in products rather than personal services.
- The small size of the SEI requires highly leveraged resources and a very effective means of technology transition.
- Technology transition requires knowledge of, and involvement with, technologies, user communities, and the transition process. While most SEI activities focus on software technology, they have a planned "side effect" of increasing SEI knowledge about user needs in specific domains.
- The SEI must balance technical depth with a broad understanding of software practice in its personnel.
- The SEI must maintain its position as an objective third party to function effectively as a center of excellence in software engineering technology.
- The SEI must act under the assumption of a DoD "no growth" strategy and plan for the expansion of resources to support the objectives of the NTP through sponsorship by those federal agencies charged with its implementation.
- The SEI must not violate contractual constraints prohibiting competition. Our approach to transition, which seeks to use the existing U.S. infrastructure as partners, helps us to avoid competition.

2.5 Conclusion

The importance of software to our national security, both in terms of defense and economic well being, has never been more clear. As articulated in this chapter, the importance of software emphasizes the importance of the SEI mission to advance the state of the practice of software engineering. The SEI is firmly committed to this mission, and has established and implemented the strategic framework described in this chapter to achieve its mission. The foundation for this strategic framework is found in our technical program, described in Chapters 3 and 4. The goal of the SEI technical program is to improve software engineering practice by maturing the process, the technology, and the profession.

In responding to the challenges discussed in this chapter, the SEI seeks to take advantage of the organizational, historical, and situational differences that distinguish it. These differences include:

- Accomplishments to date, particularly in the areas of software process modeling, real-time systems, software engineering education, computer emergency management, and software architecture.
- The SEI status as a federally funded R&D center chartered to act as an objective broker performing software engineering technology development and transition.
- The SEI association with Carnegie Mellon University and its relationship to its world-class faculty in such areas as computer science, electrical and computer engineering, and economics and business.
- The SEI sponsorship by ARPA, which provides us access to the ARPA software research community.
- The SEI charter to provide research and technology support throughout the federal government, enhancing our ability to support the objectives of defense conversion and the NTP.

The SEI has established itself as the leader in the field of software engineering. We have developed and are implementing a strategic framework for improving the state of the practice of software engineering. In supporting this framework with a strong technical program and a well-focused core competency in software technology transition, the SEI has positioned itself to respond effectively to new requirements and technologies.

Chapter 3: Table of Contents

3 Technical Focus Areas

3.1 Introduction

In this chapter we describe the Software Engineering Institute (SEI) technical program and how it will improve software engineering practice through the strategic framework.

3.1.1 Focus Area Structure

The SEI concentrates its technical effort in areas of technical focus called *focus areas.* Within each focus area, we identify and transition those technologies that we believe will most effectively mature the organizational and managerial processes, technologies, and engineering practices used by software engineering professionals. These focus areas are software process, software risk management, disciplined engineering of software-intensive systems, and trustworthy networks. Their relationship to the SEI strategic framework is shown in Figure 3-1. Within the focus areas, products are described in related clusters, which we call *product activity areas.*

The software process focus area has responsibility for maintaining competency in software process maturity modeling, definition, and measurement. The newly formed disciplined engineering of software-intensive systems focus area combines the previous focus areas of methods and tools and real-time distributed systems. This new focus area maintains competency in methods and tools for the disciplined engineering of software systems. The risk management focus area has increased its emphasis on acquisition as recommended by the Joint Advisory Committee (JAC) that assists the Advanced Research Projects Agency (ARPA) in guiding SEI activities. The new focus area in trustworthy networks acknowledges the increasing importance of trustworthy software. Its initial focus is on computer networks and their potential vulnerability to disruptive or otherwise criminal activities, which builds on the effectiveness of the SEI Computer Emergency Response Team (CERT) in countering such activities.

While all activities broadly support the strategic framework, software process and software risk management are principally directed toward maturing the process. Likewise, disciplined engineering and trustworthy networks are principally directed toward maturing the technologies and engineering practices. The activities described in Chapter 4 are concerned with transitioning more mature processes, technologies, and engineering practices into the profession.

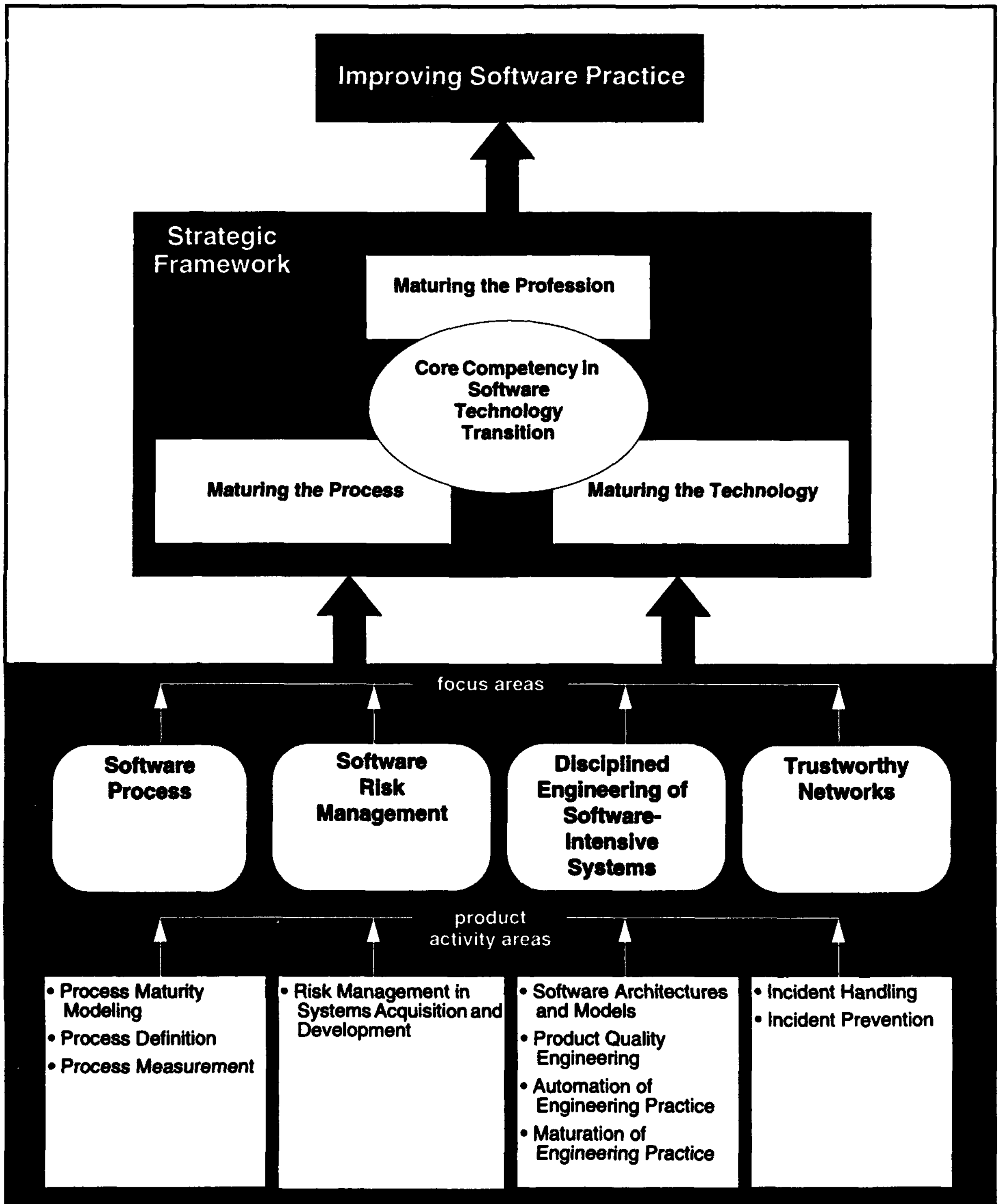


Figure 3-1: Proposed 1995 Strategic Framework, Core Competency, and Focus Areas

Chapter 3 discusses the following focus areas and product activity areas:

Chapter 3: Technical Focus Areas

Focus Areas		Product Activity Areas		Page
3.2	Software Process	3.2.4	Process Maturity Modeling	42
		3.2.5	Process Definition	48
		3.2.6	Process Measurement	53
3.3	Risk Management	3.3.4	Risk Management in Systems Acquisition and Development	72
3.4	Disciplined Engineering	3.4.4	Software Architectures and Models	97
		3.4.5	Product Quality Engineering	103
		3.4.6	Automation of Engineering Practice	108
		3.4.7	Maturation of Engineering Practice	115
3.5	Trustworthy Networks	3.5.4	Incident Handling	134
		3.5.5	Incident Prevention	136

3.1.1.1 Software Process Focus Area

Through our focus on software process ("process" for short), our objective is the maturation of the organizational and managerial processes employed by software engineering organizations. Through this focus area, we seek to define, model, and measure the maturity of these processes. The outputs have been highly visible within the software engineering community worldwide. In particular, the Capability Maturity Model (CMM) for Software, introduced by the SEI to describe the maturity of organizational and managerial processes, has been widely adopted as the basis for guiding software engineering improvement efforts, and its concepts are being incorporated in U.S. and international standards. It has also been responsible for the formation of software engineering process groups (SEPGs) in a large percentage of the Department of Defense (DoD) contractor community. Due in part to software process efforts, software process improvement network (SPIN) groups have been formed in many major U.S. metropolitan areas.

3.1.1.2 Software Risk Management Focus Area

Through our focus on software risk management ("risk management" for short), we provide a systematic and structured process, supported by methods and tools, for identifying, analyzing, and mitigating the uncertainties encountered within a specific software engineering effort. We are focusing on risk because many of the most serious issues encountered in systems acquisition were the result of risks that remained unrecognized until they had already created serious consequences. Work in risk management is founded on the belief that (1) structured techniques, even quite simple ones, could be effective in identifying and quantifying risk; and (2) techniques existed to mitigate risk. The thrust is thus to identify and quantify risk. As we

have gained experience through more than 30 software risk evaluations (SREs), we have broadened our work to include team risk management and risk modeling and measurement activities.

3.1.1.3 Disciplined Engineering of Software-Intensive Systems Focus Area

Through our focus on the disciplined engineering of software-intensive systems ("disciplined engineering" for short), our objective is the identification and validation of disciplined engineering practices, techniques, and technologies and their transition into software practice. Efforts are focused on two aspects of disciplined engineering: (1) enabling practitioners to share common views and models of architectures for similar systems; and (2) identifying technologies and engineering processes for defining, analyzing, predicting, and controlling performance, reliability, interoperability, and other quality attributes of software systems.

3.1.1.4 Trustworthy Networks Focus Area

Through our focus on trustworthy networks, we are concerned with ensuring that computer networks, especially the Internet and eventually the National Information Infrastructure (NII), can be trusted to maintain their own integrity and security and the integrity of the data that they transport or store. This work draws on the experiences gained through the SEI CERT since its formation by ARPA following the seriously disrupting Morris Worm incident of 1988. In addition to responding to incidents through CERT, this focus area also seeks to mature network security technologies and practices.

3.1.2 Extending the CMM for Software

The ultimate purpose of the CMM is to permit assessment of the capability of a software development organization to develop high quality products. The CMM also creates a basis and guide for improvement. As it has thus far evolved, the CMM principally addresses organizational and managerial processes and practices. While these are perhaps the most important determinants of capability for relatively immature organizations, they alone do not determine overall capability. Other factors such as an organization's human resources, the maturity of its technology and engineering practices, and its intellectual property and acquisition practices and processes may need to be included.

Initially, we are developing separate maturity models for those determinants of capability associated with human resources and technology through the People Management Capability Maturity Model (PMCMM) and the Engineering Maturity Model (EMM), respectively. The PMCMM is described in Section 3.2, and the EMM is described in Section 3.4. Once these models have been separately developed and validated, we intend to incorporate them into the CMM.

We envision the CMM as a model that can be extended to encompass all factors that determine capability. Our ultimate objective is to address all determinants in one single, extended CMM.

3.1.3 Other Maturity Models

The CMM has been very successful. Consequently, others are developing similar models to address capability in other domains. Examples are the Systems Engineering Capability Maturity Model (SECMM) of the maturity of systems engineering and the Systems Acquisition Capability Maturity Model (SACMM) of acquisition processes. Both efforts are being managed/coordinated by the SEI. The SECMM effort, an initiative of industry, is described in Section 3.2. The SACMM effort, an initiative of government, is described in Section 3.3.

3.1.4 Advisory Boards

Each focus area maintains or is establishing advisory boards. These boards meet periodically (usually quarterly) to review the activities and plans of their respective focus areas. The boards are made up of key people from government, industry, and academia. For more information on these boards, refer to the specific focus area.

3.1.5 Measures of Success

Evaluations are provided annually by our technical objectives and plans (TO&P) customers via "sponsor feedback forms." The most desirable measures of success would be those that bear direct relationship to key business performance indices like Return on Investment (ROI) and productivity. These, however, cannot be obtained immediately where significant time passes between the initiation of process improvements and the effects of those improvements. This is the case, for example, in instituting the changes required to move to the next higher CMM maturity level. Initially, one can measure success by the extent to which changes are initiated, the number of people trained, the numbers of assessments or evaluations performed, etc. Later, sometimes several years later, enough effects on business indices of the improvement efforts will have accumulated to permit success to be quantified. This is beginning to be the case with the measures of success data that have been acquired to date.

Wohlwend and Rosenbaum [Wohlwend 93] of Schlumberger Laboratory reported experiences with CMM-derived improvements from 1989 to 1993. A Schlumberger group that works on complex embedded real-time systems reported that it was able to reduce the number of validation cycles from 34 to 15. This, in turn, reduced time to market. The group credits this improvement to the introduction of requirements management, a level 2 KPA.

Wohlwend and Rosenbaum also reported that one engineering group began concentrating on process improvements in mid-1990 and improved on-schedule completion from 51 percent in 1990 to 89 percent in 1991 and to 94 percent in 1992. The group credits this to improvement in initial project planning, another level 2 KPA. For each level 2 KPA introduced, Wohlwend and Rosenbaum describe similarly impressive improvements.

Lipke and Butler [Lipke 92] of the Oklahoma Air Logistics Center (OC-ALC) described efforts that began in 1989 to introduce SEI process improvement into the Aircraft Software Division (LAS) of OC-ALC. To guide their efforts, LAS established a Quality Management Steering

Team and an SEPG. The LAS SEPG wrote an action plan for improvement that is periodically updated and revised. The purpose of the action plan is to set improvement goals; to discuss how improvement efforts will be measured; and to provide a template for planning, developing, and implementing improvements. As of November 1992, LAS had introduced 44 improvements and had gathered ROI data on 18 of them. For those 18, $2.935 million was returned by an investment of $462,100, resulting in an ROI of 6.31.

Putnam [Putnam 93] reported results of moving up the SEI CMM scale from a high level 1 to a low level 3 from1988 to 1992 for one system done at a central design agency within the DoD. Figure 3-2, taken from Putnam's report, shows ratios of improvement from 1.7 to 5.7 in measures such as mean time to defect (MTTD), peak staffing, cost, effort, and time. These are consistent with previously published improvements at Hughes [Humphrey 91], Raytheon [Dion 92], and Hewlett-Packard [Grady 89].

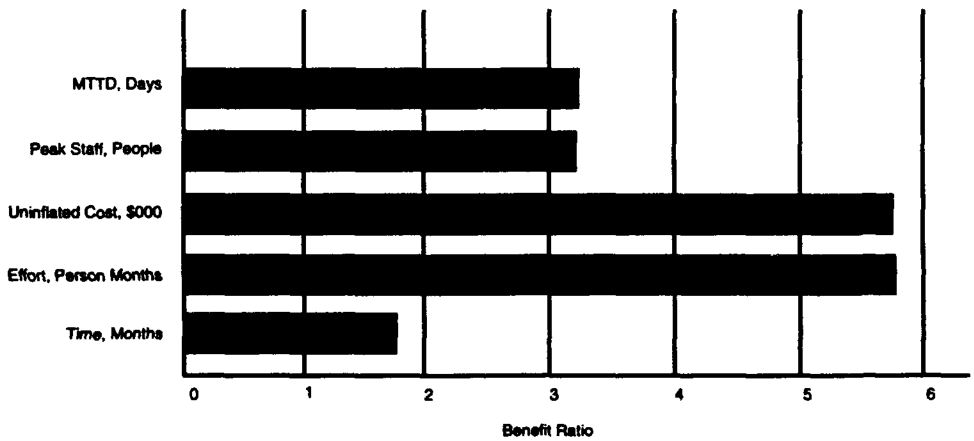


Figure 3-2: Benefits of Moving from Level 1 to Level 3

During 1994, the SEI was able to analyze software improvement data provided to it by a number of different corporations. While there were no consistent measures across corporations, it was possible to quantify and compare relative improvement. For organizations that had been engaged in process improvement for periods of three years or more, an increase in ROI of 4:1 to 8.8:1was achieved. This effort is continuing, and the SEI seeks to take a leadership role in collecting, aggregating, and analyzing data that can track the relationship between SPI activities (including those based on the CMM) and resultant ROI.

3.2 Software Process

Large numbers of software development organizations in the United States continue to have an ad hoc, crisis-driven process that often results in projects producing poor-quality systems that are chronically late and over budget. Indeed most software organizations are at the lowest level of software process maturity when appraised against the SEI CMM for Software (based on 261 SEI-assisted, vendor, and self assessments as of March 1994; see Figure 3-3).

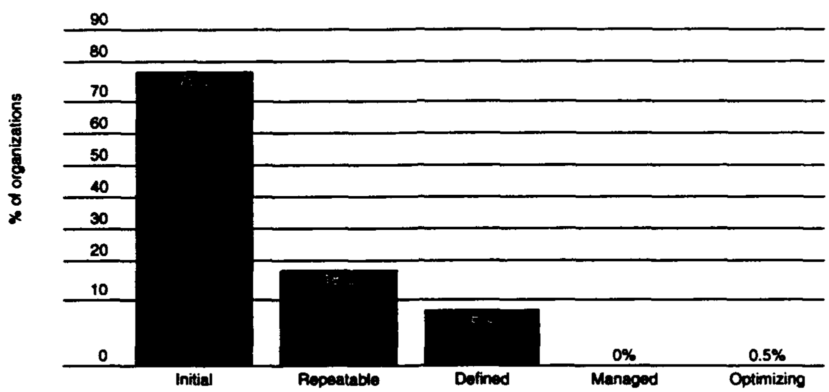


Figure 3-3: Organization Maturity Profile

However, organizations that have established long-term software process improvement (SPI) efforts report evidence that they are beginning to show movement toward higher levels of maturity, as well as business-related improvements. Figure 3-4 reflects data reported to the SEI on the maturity improvements of organizations that have been reassessed. Fully 83 percent of the 23 reporting organizations achieved capability maturity improvements between assessments. (Business-related results are discussed in detail in Section 3.2.1.1.)

Lower maturity, crisis-driven software environments are characterized by:

- Unpredictable, inconsistent performance and quality.
- An inability to successfully implement and sustain new technologies and methods.
- Strong dependence on a few highly competent individuals.
- A focus on firefighting.
- High levels of frustration and adversarial relationships across disciplines.
- Predominantly schedule-driven environments.

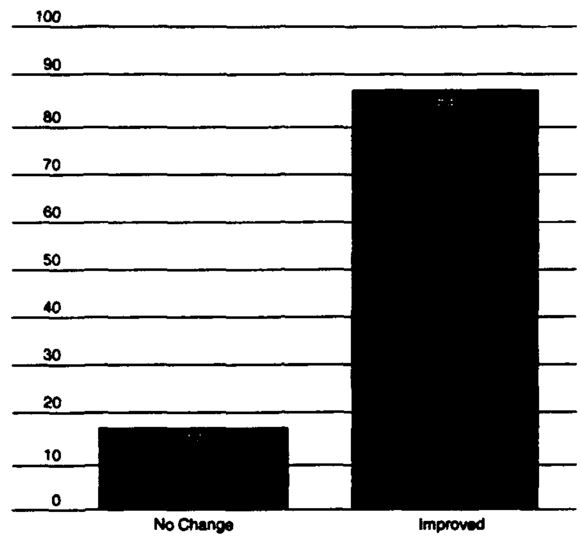


Figure 3-4: Change in Maturity Level Between Reassessments

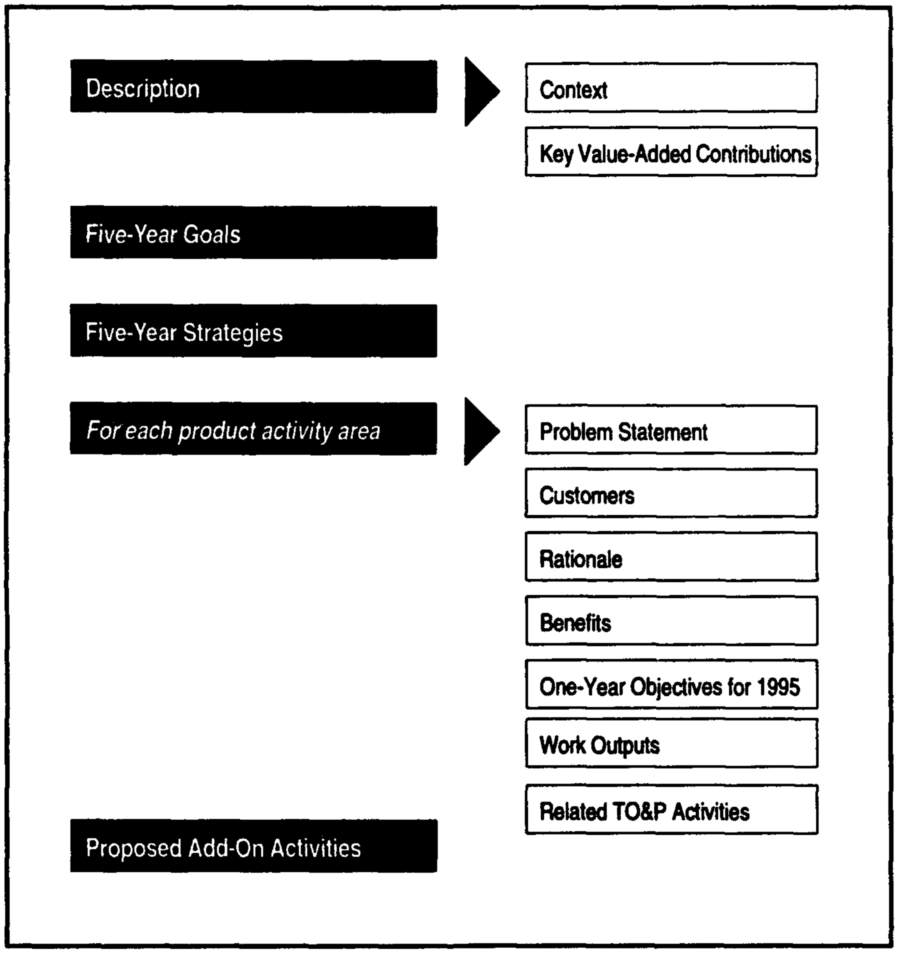
Too many software organizations have long relied on individual talent, which is in short supply, to produce acceptable results. Unfortunately, the only way these results can be repeated is to assign the same individuals to the next project. In such environments there is no institutionalized capability for meeting projected targets for cost, schedule, quality, and functionality. Executives in such organizations typically complain that they have little visibility into the software development process and that they are unable to make accurate projections of performance and costs. Without visibility and predictability, executives are unable to exercise management oversight or to make sound business decisions regarding projects.

To improve the state of the practice of software engineering in the United States, software-producing organizations must establish an organizational capability (rather than be dependent on individuals) for developing software based on sound management practices that support a disciplined, defined, and measured software engineering process. They must be able to execute this defined engineering process consistently across all projects in the organization, rather than have only a few successful projects, with others missing the objectives of cost, schedule, quality, and function. Furthermore, organizations must be able to learn from their experience to improve their capability.

Establishing an organizational capability for developing software also entails defining and implementing software measurement practices. Measurement, and the ability to see and understand progress, are closely related—a developer can measure only what is visible in a process, and measurement helps to increase visibility. The CMM can serve as a guide for determining what to measure first and how to plan an increasingly comprehensive measurement program. For example, measures at level 2 focus primarily on project planning, management, and tracking, while measures at level 3 become increasingly directed toward intermediate and

final products. Measures at level 4 capture characteristics of the development process itself to allow control of the individual activities of the process. At level 5, processes are mature enough and managed carefully enough to permit measurement to provide feedback for dynamically changing processes across multiple projects.

The sections for this focus area are:



3.2.1 Description

3.2.1.1 Context

The SEI maintains a core competence in process maturity modeling, definition, and measurement to be a primary driving force in maturing and improving software management and organizational processes. The products and services provided by this focus area are all based on the CMM, which also supports advancement in the other technology focus areas and the maturing bases of the strategic framework (see Figure 3-1). Additionally, the transition of the CMM into practice supports the SEI competence in technology transition and is our key contribution to maturing the practice of software engineering. This transition is enabled by the

products and services provided by the process focus area in support of continuous process improvement.

The CMM has been stabilized for 1994-1995 and is under change control. Efforts in areas of community awareness of the CMM have continued to be a high priority. Involvement in work to influence international standards toward U.S. positions has become very intense and a top priority. The modeling efforts have been expanded to integrate Systems Engineering and PMCMMs with the CMM. This has led to initial work on an integrating architecture for the maturity modeling efforts across the SEI, based on the draft SEI-proposed international standards architecture for such models.

The CMM-Based Appraisal (CBA) effort was formed by merging two previously separate efforts: software process assessments and SCEs, which share common elements. This effort has produced new products for assessments and evaluations that are based on the latest version of the CMM. The integration of these two appraisal types is also leading to development of a common appraisal architecture: the Common Rating Framework (CRF). A CRF for baselining appraisals will lead to more consistent results across appraisal types. This CRF is being defined in 1994.

In 1995, integration activities will be addressed for the empirical methods, software process definition, and software process measurement activities. The feedback from our customer community is that they often align their SPI efforts to address the various activity areas in the process focus area. This often leads to multiple, uncoordinated improvement actions. The focus of our activities will be to address how we can provide a more integrated approach to SPI efforts and build integrated process products and services.

Additional work to identify and report the results of SPI efforts, some of which has been completed and some of which continues, responds to some of the most prevalent questions from our community, and is reported in quantified business terms relating to cost, schedule, productivity, quality, and ROI in SPI (summarized in Figure 3-5).

3.2.1.2 Key Value-Added Contributions

Through our work in process, we have established a community-owned, de facto standard model of organizational and management discipline and process maturity that envisions a culture of software engineering excellence. We maintain stewardship over the model on behalf of the community to ensure its continual improvement, reflecting future best practices and emerging states of the art. In addition, transition of the model into practice in some leading-edge customer organizations has been accomplished by applying the products and services of the process focus area.

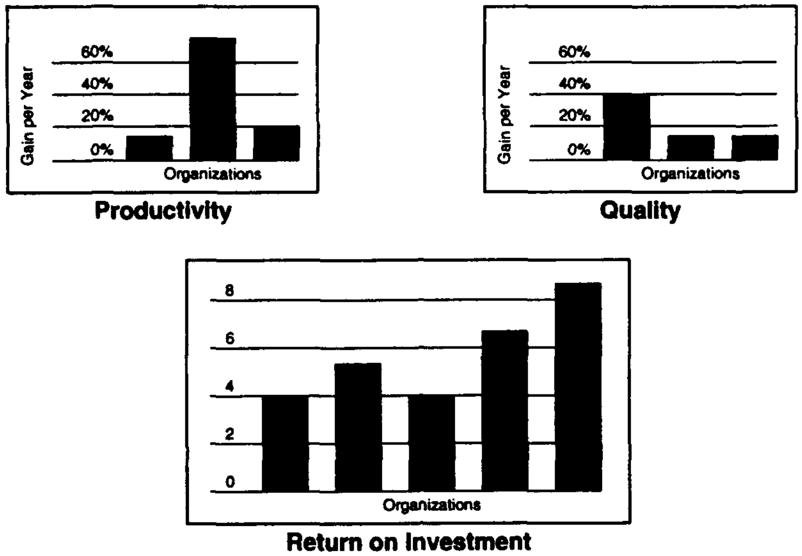


Figure 3-5: Preliminary Results of SPI*

* These charts illustrate some of the gains reported by organizations using CMM-based SPI. Each bar represents a data point provided by one organization. The data on the three charts do not necessarily all come from the same organizations. The first two charts show yearly productivity and quality gains for improvement efforts. Among these organizations, typical gains are 10% per year in productivity and quality, while some have experienced much greater improvements. The third chart shows the ratio of savings to costs for the process improvement effort. These organizations, all of which have been engaged in SPI for at least three years, are showing very impressive ratios in the 4:1 to 8.8:1 range.

We are increasing our competence in SPI with the objective of being able to include SPI in our core competence in process maturity modeling, definition, and measurement. Using the model as a base and coordinating with various other sources, organizations are educated and trained in assessing their current state, planning for improvement, defining improved processes, and measuring those processes to determine business results associated with the improvements. As the products that support this training become mature, they are transitioned to leverage the proven techniques.

The SEI is a trusted safe haven for proprietary data reflecting the results of SPI. Participants supply the results of assessments they have undergone as well as the benefits of investing in SPI. We report these results to the software community so organizations can benchmark their own state and progress and make decisions on areas of SPI investments. See Figure 3-5. Additionally, as a neutral player in the community, the SEI process focus area is looked to for coordination and assistance across the community to support individual organizations' development of SEPGs and networking through SPINs.

3.2.2 Five-Year Goals

The process focus area has four long-range goals, described in detail below.

1. Establish a shared model of process maturity that envisions a culture of disciplined software engineering excellence.

The primary goal is to transition the CMM into the state of the practice of software engineering to improve the maturity of software development organizations as defined in the CMM. The products and service facilitate and enable such a transition.

2. Establish continuous process improvement in the manner in which software engineering is practiced for customers to meet their business objectives.

The second goal is to transition an integrated approach to SPI to the software community, as illustrated in Figure 3-6. An organization should be able to look to the SEI for a suite of products and services to determine where and how it should start and continue along the path of continuous process improvement for software development. The ultimate goal here is that by 1999 all major commercial software producing organizations, as well as those that serve the DoD community, will have instituted continuous SPI programs.

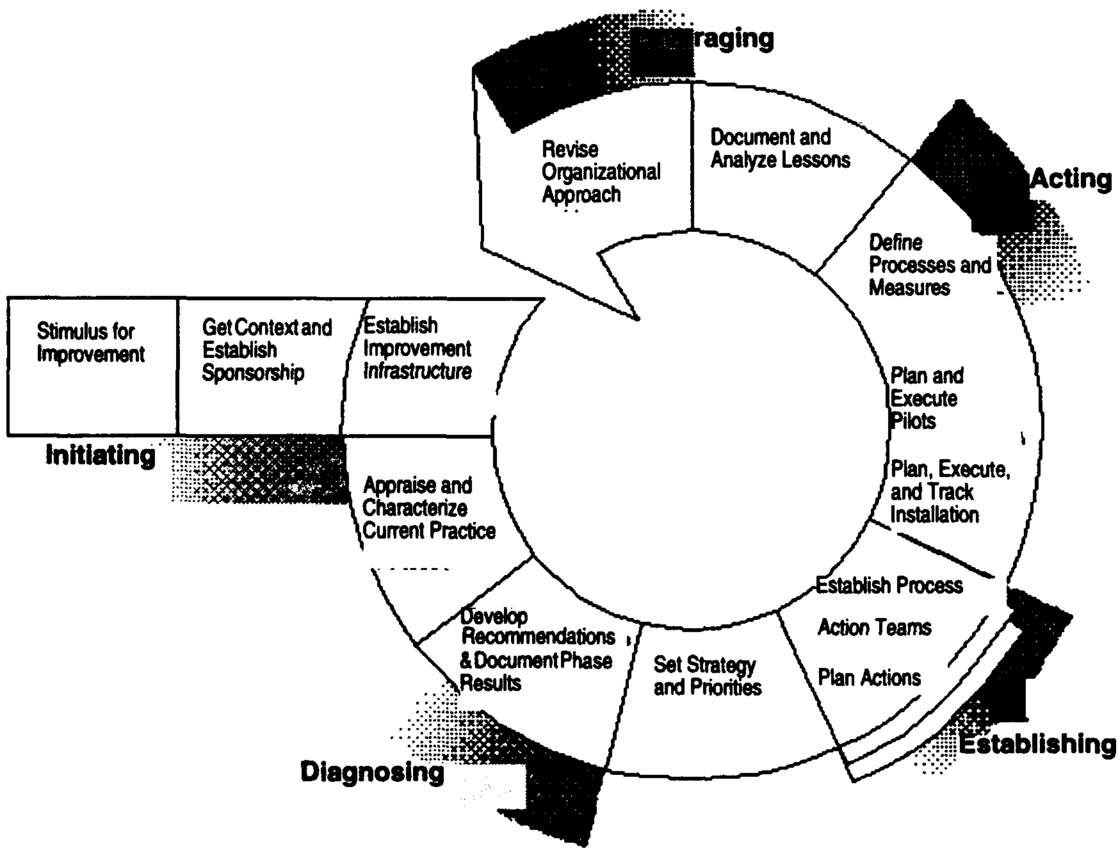


Figure 3-6: An Integrated Approach to SPI

3. Develop and deliver products that enable the transition of the model and continuous process improvement.

The third goal is to provide products that are integrated with each other and with the CMM. These products support the activities of an integrated approach to SPI as shown in Figure 3-6. The ultimate goal here is to see a strong commercial infrastructure to support SPI developed within the United States, by 1999, to provide inter-organization communication about lessons learned in process improvement and wide competitive choices for satisfying the needs of the software development community. A sufficient cadre of trainers and consultants will be in place for educating customers about organizational change and process improvement and sustaining customer improvements. Efforts to promote a commercial process improvement industry, with quality standards for practitioners through such organizations as the International Standards Organization (ISO), for example, with ISO9000 and SPICE (Software Process Improvement Capability dEtermination), will continue, and we will continue to work to ensure that the SEI and U.S. perspectives for SPI are addressed in such standards. University programs will also teach SPI, including personal and team software processes.

Process improvement programs, based on the SEI approach, are planned to dramatically improve the quality and reduce the costs and schedule slippage of software developed by and for government and commercial software producing and procuring agencies. We expect SPIN groups to continue to increase internationally (see Figure 3-7). Total membership in the 19 SPINs is approximately 3,400 individuals. We also expect each SEPG national meeting will continue to attract in excess of 800 participants (see Figure 3-8).

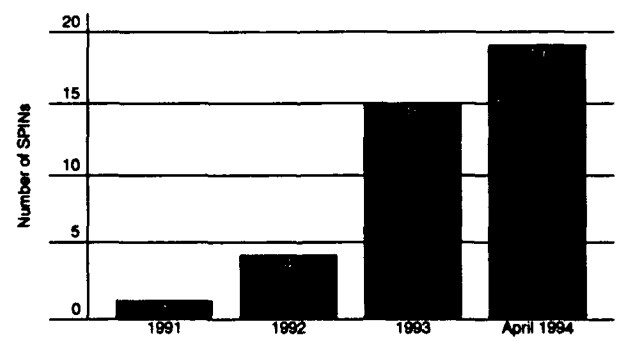


Figure 3-7: SPIN Group Growth

4. Assess and communicate the results of continuous process improvement.

The SEI is viewed as a safe haven for collecting, analyzing, and reporting on data from the community on the results of SPI. These results are needed by the community to evaluate and justify the efforts of organizations to continuously improve and to identify business results associated with SPI.

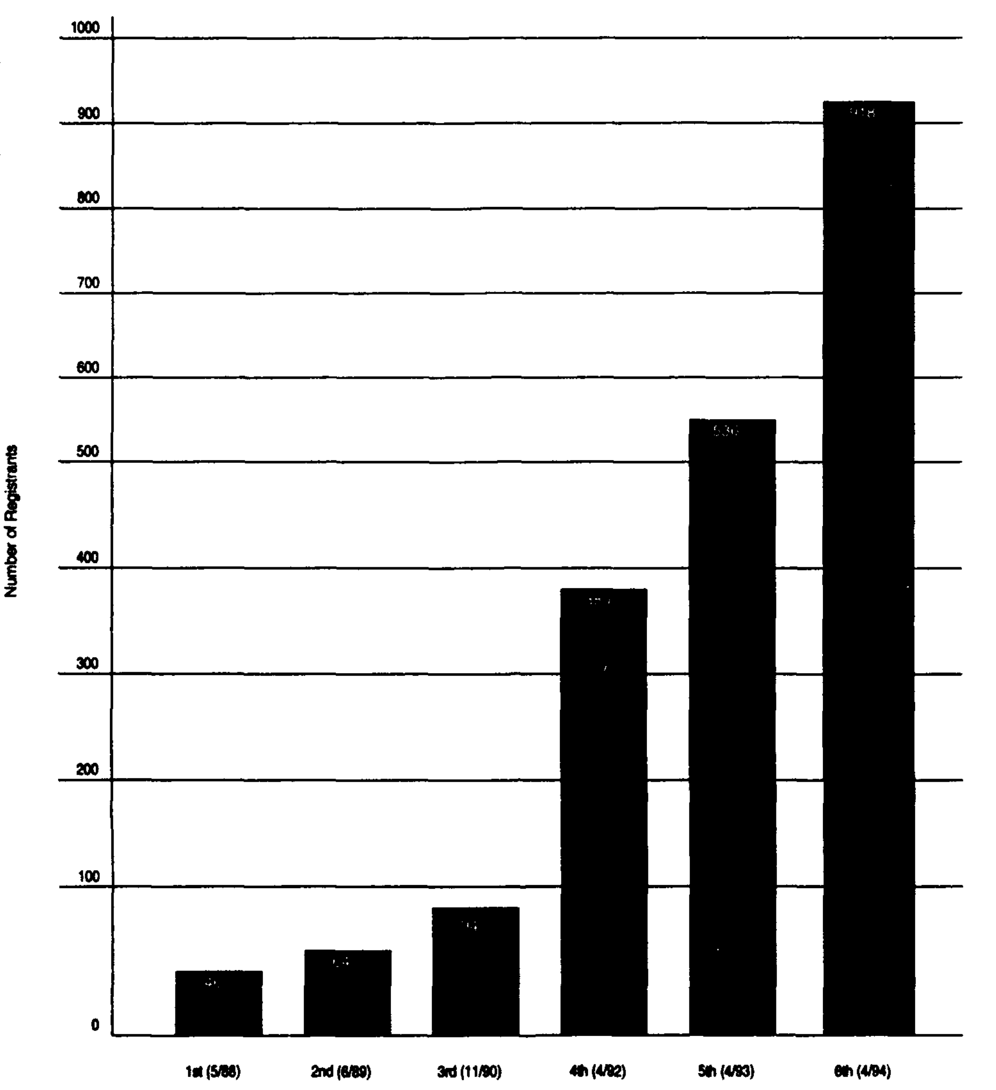


Figure 3-8: SEPG National Meeting Attendance

This fourth goal is intended to lead a national commitment to improvement and a service business to support it, so that by 1998, 80 percent of all defense contracting sites with more than 50 software engineers will have active process improvement programs, and 50 percent will have advanced one level from their initially measured maturity level. We expect that by 1999 this goal will address other government software producing agencies; and by 2000, commercial software development organizations. Productivity, quality, and process measures will be routinely collected and used to improve the software process in those organizations. A sufficient number of contractors with a defined, measured, and managed software process will exist by 1999, so that government agencies will not need to use contractors with weaker software

process capability for software-intensive systems costing $10M or more. Further, by 1999 all software systems procured from contractors with high process maturity (Managed or Optimized levels) should be delivered with fewer than one-tenth of a defect per thousand source lines of code.

Achieving these goals should make U.S. software developers among the most competitive in the world in terms of cost, quality, and time to market. This focus on process will generate requirements for process technologies that can be incorporated into software engineering environment (SEEs).

3.2.3 Five-Year Strategies

Achieving these goals will require maintenance and evolution of the CMM. The CMM, developed at the SEI with much community participation, describes five stages through which software organizations must pass to achieve a sustainable state of continuous process improvement. This five-stage model provides software organizations with guidance in planning a long-term process improvement program, in addition to assistance in setting priorities for near-term improvement activities.

The process focus area has five long-term strategies, described below.

1. Maintain and evolve the CMM.

Our strategy is to maintain the CMM as a community-owned model for which the SEI provides stewardship until such time as a standards organization, e.g., ISO, adopts the CMM as the preferred standard. The CMM is constantly subjected to national and international review, has been stabilized at Version 1.1, and has been accepted as the de facto standard for implementing SPI activities. It is now common practice in technical literature to simply say "SEI CMM level 1" without explanation.

Future versions of the CMM must address new and changed requirements from the software engineering community. New key process areas (KPAs) need to be considered, further refinement of higher level maturity KPAs is needed, and change requests, such as having KPAs span levels, need to be investigated and incorporated into the CMM.

2. Integrate CMM concepts into ISO standards.

A wide range of American software developers have urged the SEI to integrate CMM concepts into ISO standards. We continue to pursue these requirements. For example, the SEI is leading the U.S. delegation working with the ISO SPICE Project, which is working on international standards for SPI and capability determination. We will actively participate in revision efforts of ISO 9000-3.

3. Extend the CMM.

The SEI is providing coordination across a large community of contributors and interested parties in the SECMM. This model will extend the CMM concepts and approaches upstream to

the suppliers of software requirements. It will also provide appraisal techniques to systems engineering much like what has been provided to software organizations. Additionally, an SEI proposed maturity model architecture, which allows KPAs to span maturity levels, is being piloted with the SECMM for ISO SPICE.

A People Management Capability Model (PMCMM) has been drafted to complement and integrate with the CMM. This model provides five stages of maturity through which individuals and organizations pass to improve the multiple aspects of human resources skills and operations. The development of the PMCMM involves several contributors from the software community.

4. *Transition the CMM into the state of the practice.*

a. Provide and transition products to assess the state of the practice.

To use the CMM for guiding process improvement activities such as those we have conducted at various agencies, including the Army Materiel Command (AMC) and the Air Force Materiel Command (AFMC), we will continue to provide software organizations with methods to reliably assess the maturity of their own development and maintenance processes. Concurrently, we must provide acquisition organizations such as AMC, AFMC, and the Naval Air Systems Command with the ability to reliably evaluate the capability of their contractors.

For a variety of strategic reasons, the SEI is transitioning to third parties the capability to train and execute these diagnostic methods. The software process assessment technique has been upgraded to CMM Version 1.1, and training is available to individuals who desire to lead assessment teams, be members of assessment teams, or perform internal assessments. Agreements with companies and agencies that were licensed in the pre-CMM assessment method will expire at the end of 1994, while individuals will be authorized to conduct CMM-based assessments during 1994 and beyond.

In 1993, we began to transfer the ability to train capability evaluation teams to training sources inside the DoD such as the Defense Systems Management College (DSMC). At this writing, cooperative research and development agreements have been signed between the SEI and several commercial firms to further expand the availability of training for capability evaluation teams. Although the SEI intends to provide ongoing control of core elements of its appraisal methods such as the CMM and quality oversight, further transition of the instruments of appraisals may be pursued if the initial commercialization activities are successful.

To affect growth in the national software capability, these diagnostic methods must be coupled with the ability to plan and implement process improvement programs tailored to the maturity level and specific problems of software organizations throughout government agencies and industry. Since these evaluation and assessment methods are rapidly growing in use and in importance in the software community, it is essential that qualification be established during this strategic period. It is expected that a qualification program will be fully established by 1995 for

both CBA for Software Capability Evaluation (SCE) evaluators and CBA for internal process improvement (CBA-IPI) assessors.

b. Transition process definition capabilities.

After an assessment of process capability, an organization has great need for a defined process upon which to structure process improvement efforts. Particularly for level 1 organizations, there is an uncertainty of the relative priorities of the activities necessary to start a process improvement effort. Several products related to process definition are essential to maintaining process improvement momentum:

- A defined process for process definition.
- Examples of good software processes.
- A library of proven process assets.
- Procedures for establishing requirements for a process.
- A representation language for process definition.
- Procedures for designing a process.
- Procedures for implementing and installing a defined process.
- Procedures for evaluating a defined process against a standard.
- Procedures for measuring the effectiveness of a defined process.
- Education and training on process definition.

Collecting technology for these products, creating the products, and getting them into widespread use is underway and will be a major activity during the coming five-year period.

c. Provide products and transition software measurement capabilities.

In addition to needing defined software processes, an organization also needs measurements that can reflect maturation of these processes. Such measurements are necessary for sustaining process improvement programs, communicating results of process improvement, and for providing feedback on the potential for additional areas of improvement. A software measurement program is also part of the process for managing software development. Several products related to process measurement are essential for maintaining process improvement momentum and for assisting in management of software development:

- A baseline set of core measurements.
- Definitions of the components of the core measurements.
- Techniques for estimating and predicting trends and results.
- Education and training on measurements and estimating.
- Example applications of measurements and analysis of the resulting information.
- Benchmark data reflecting the results of SPI.
- Assistance in installing a measurement program.
- A reference handbook for software measurement.

Providing support for the measurement programs of initiating organizations and sustaining the collection and analysis of the resulting data will be a major activity during the coming five-year period.

5. Strengthen the community infrastructure.

The SEI supports the development of SPI programs throughout government and industry sectors. To introduce these programs broadly in both sectors, we must continue to support the formation and development of SEPGs and to educate and train the members of SEPGs in the skills and tactics that have proven to be successful in executing improvement programs. The SEI supports these SEPGs by providing them with specific methods and training designed to enhance SPIs. In executing an improvement action plan, SEPGs need methods for defining and measuring software processes. We need to continue to develop, evolve, and pilot these methods and determine that they can be installed in the context of an improvement program. While we have conducted pilots such as those at the Naval Air Warfare Center (NAWC) and AMC, we must continue to develop and transition courses based on our direct experiences to prepare others to incorporate these methods in their improvement programs.

Organizations ask how they can quantify improvements before they introduce new SPI approaches. The SEI began to provide initial reports of the results of SPI in 1993. This work, using various sources of data and various levels of reporting, will continue to be made available to the software community to support sustaining SPI efforts.

The process focus area has several advisory boards or steering committees with skilled and qualified representatives from academic, industry, and government sectors. These groups will continue to provide advice throughout this strategic period. It is our intent to have minimal duplication of focus between these groups. (See Appendix B for a description of these groups.)

3.2.4 Process Maturity Modeling

This section discusses the product activity area related to CMM, PMCMM, and SECMM.

3.2.4.1 Problem Statement

Software organizations need guidance on how to recognize where there are deficiencies, as well as strengths, in their organizational and managerial processes. A model of disciplined software engineering practices is required to support the quality improvement efforts of organizations.

Additionally, lack of guidance for process improvement within the systems engineering discipline has led to a recognition of problems similar to those in the software engineering field. Systems engineering needs guidance in meeting its potential as an integrating discipline and, as an upstream supplier to software engineering, systems engineering shortcomings have an impact on software engineering organizations.

Finally, given the recognition of the problems and a model to help guide improvements, there is a need to develop and transition state-of-the-art process maturity models, languages, auto-

mation, and technology into practice in the software engineering community. Advanced support for continuous improvement and reengineering of software processes is needed to (1) identify requirements for, and (2) conduct prototype exploration of, advanced technology and techniques. Research collaboration agreements will be needed to leverage the efforts of SEI and external researchers and developers to more effectively progress in meeting the current and future needs of the community.

3.2.4.2 Customers

Both software and systems development communities will benefit from the process maturity modeling efforts of the SEI, especially those organizations whose products are in response to contracts from government agencies or industrial partners and those that are market/customer-driven. Customers will also include the product acquisition community, government program offices, and industrial firms. These latter include commercial organizations, multinational firms, standards communities, and education/training businesses.

3.2.4.3 Rationale

There has been an overwhelming response from the software community to the availability of the CMM. Customers and suppliers of SPI regularly reference the model, its maturity levels, KPAs, and practices. The CMM has become a de facto standard internationally, and has much support in the global community to be the driving force and cornerstone in various ISO software standardization efforts.

Multiple views are taken by the community in leveraging and transitioning the model into practice. The CMM provides the community and individual organizations with:

- A vision of a culture of software engineering excellence.
- A roadmap for organizational improvement toward the vision.
- A community-owned and shared model of organizational and managerial discipline and process maturity that facilitates both communications and continuous process improvement.
- A need for the identification and sharing of "best practices" within the software engineering community.
- Improved software quality, schedule, and cost results, when transitioned into the practice of software engineering.

There is a growing emphasis on appraisals and SPI efforts based on the CMM. These are the vehicles that enable the transition of the model into the state of the practice of software engineering. The business results of SPI are being gathered and reported (see Figure 3-9), and they support the value of the model-based approach to SPI. Efforts are needed to validate the model (see Figure 3-10) as well as its constituent parts, the KPAs.

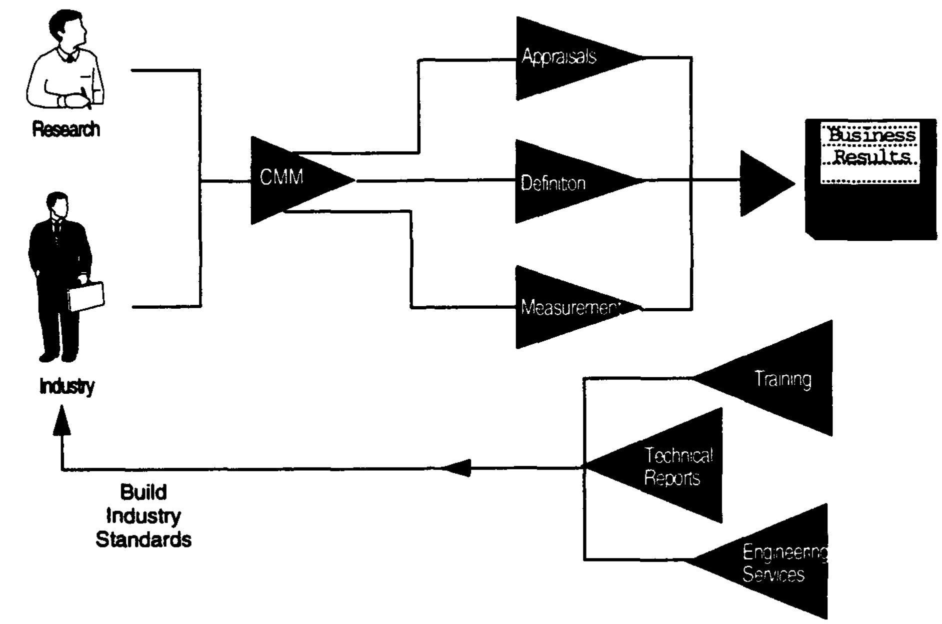


Figure 3-9: Business Results of SPI and Validating the Model

SEI field work over the years has revealed that the findings of software engineering appraisals frequently have a root cause in the systems engineering portion of a project or organization. The National Council on Systems Engineering (NCOSE), a professional society of systems

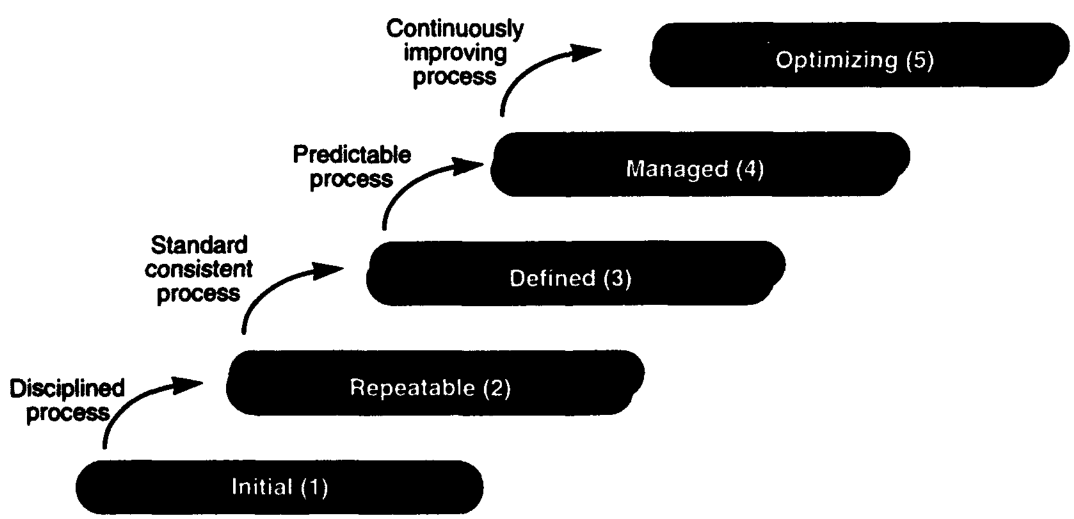


Figure 3-10: Capability Maturity Model Concepts

engineers, began a volunteer-based project to build a model/appraisal method in 1992. Based on the SEI core competence in process maturity modeling, the process focus area was approached to coordinate this national effort. Leading aerospace companies, as well as others, are investing one full-time equivalent contributor plus travel to support development of a prototype model in 1994. Other non-SEI appraisal methods, such as the Software Development Capability Evaluation, include addressing systems engineering to support evaluation of overall contractor capability. The SEI has frequently received this same requirement and is now in a position to assist the community in addressing it.

Additional rationales:

- Forms the basis for appraisals, often the first critical step in an improvement process.
- Forms the basis for data on state of the practice, ROI, etc.
- Is a unifying concept for definition and improvement.
- Forms the basis for international standards, which can give competitive advantage to U.S. firms.
- Provides models extensible to other domains, such as SECMM and PMCMM.

3.2.4.4 Benefits

The current version of the CMM provides the basis for continuous SPI. The primary benefit to the software community is realized when the model is transitioned, via multiple transition enablers, into the state of the practice in organizations. Business results of SPI, based on the CMM, have been publicly reported by a few organizations and are continuing to be gathered, analyzed, and now reported by the SEI (see Figure 3-9). The business results reflect increasing product development capabilities and predictability in organizations investing in SPI. The CMM-based approach strengthens an organization's ability to communicate, improve, and measure its effectiveness.

A CMM-based effort integrates well with the other Total Quality Management initiatives of organizations. It enhances those initiatives in software by the nature of its software-specific orientation. It reinforces and enhances software management process improvement.

As the CMM evolves, it will address higher levels of maturity more completely. It will reflect more mature best practices and state-of-the-art practices over time. In addition, the CMM is an additive model and can evolve by integrating with other models, e.g., the Systems Engineering CMM, the PMCMM, an SACMM, and an Engineering/Technology Maturity Model.

As the basis for continuous SPI, the CMM also integrates the other products and services provided by the process focus area. Appraisals used for IPI and for SCE for subcontractor selection are tied to the CMM. As such, and through development of the CRF, these appraisals will produce consistent and repeatable findings and ratings. The follow-on activities of SPI (action planning, process definition, and process measurement) are also tied to the CMM. This complete use of the model results in commonality of language and vision in SPI, as well as a roadmap to realizing improved organizational process maturity.

In the systems engineering modeling effort, it is anticipated that similar benefits can be realized in that community. The SECMM will provide guidance for improving systems engineering processes and potentially high leverage in supporting DoD initiatives toward using commercial standards and products. As with the CMM, this model will provide a shared vision of systems engineering excellence for the systems and software engineering communities (see Figure 3-11).

Overall strategy: 1) Incorporate innovations; 2) Integrate with emerging standards

Key Items	State of practice as of:			Impact/Metrics
	Today	+2 Years	+4 Years	
Identify, define, extend, and evolve a maturity model	• small part of the community knows CMM	• wide impact • further address levels 4&5 • integrate with SECMM	• integrated with ISO	• number of customers/ vendors using • increased maturity • institutionalized
Support and transition SPI	• task, schedule, and product focus	• increased focus on process	• process metrics	• number of customers using • increased maturity • institutionalized
Gather and transition best practices	• a few leading edge practitioners	• levels 4&5 • best practices documented	• become state of the practice	• improved maturity levels

Figure 3-11: Trends in Capability Maturity Modeling (CMM)

The SECMM is compatible with and will be integrated with the CMM and PMCMM. It is also providing us with the opportunity to pilot the modeling architecture that the SEI has proposed to ISO SPICE and insight into how that change in architecture will enhance version 2.0 of the CMM.

Finally, the SECMM will provide a basis and integrating framework for appraisal and process improvement efforts in the systems engineering community much as the CMM has done for the software community. It will provide a reference model for assessing current practices; for performing supplier selection; for planning, implementing, and measuring process improvement efforts; and for determining the business results of such efforts (see Figure 3-12).

Key Items	State of practice as of:			Impact/Metrics
	Today	+2 Years	+4 Years	
Develop SECMM compatible with CMM for Software	• small part of systems engineering community involved in process improvement • individual organizations have built models without compatibility considered	• focused application of SECMM-based improvement • expansion to cover integrated process and product development • integrated improvement efforts between systems engineering/ software engineering	• widespread application • integration with ISO 9000	• use by both market-driven and contract-driven customers • increased maturity • level of institutionalization
Develop appraisal method compatible with CBA	• NCOSE Capability Assessment Working Group method based on CMU/SEI-87-TR-23 questionnaire	• integrated engineering appraisals	• multidiscipline process capability appraisals • multidimensional capability appraisals	• use by both market-driven and contract-driven customers • number of dimensions "appraisable" • number of disciplines "appraisable"

Figure 3-12: Trends in Systems Engineering Capability Maturity Modeling (SECMM)

3.2.4.5 One-Year Objectives for 1995

The objectives for the CMM are included in the baseline core area.

1. Maintain the CMM.

This objective includes continuous maintenance of CMM Version 1.1, which consists of processing change requests, answering questions from the community, and delivering technical reports as appropriate. Introductory and advanced training on the CMM is another activity related to maintaining the CMM.

2. Evolve the CMM.

Initial work on development of CMM Version 2.0 will start late in 1994 and continue throughout 1995. This work will include enhancing levels 4&5 in the CMM, addition of other potential maturity aspects, e.g., risk, reengineering, and reuse, and investigation and incorporation of best practices. Much integration of this work with and across the community will be undertaken.

3. Integrate CMM concepts into ISO standards.

The ongoing leadership and coordination roles of the process focus area in SPICE standardization will continue in 1995, leading to technical reports and completed field experiences in 1996. Continuing involvement in other standards activities, including ISO-9001 and 9000-3

major revisions, will be a key related piece of work for our focus on process maturity modeling. However, this baseline effort is limited to that which is essential; we will be seeking TO&P funds to support broader participation. (See Section 3.2.4.7.)

3.2.4.6 Work Outputs

ISO SPICE Technical Reports (1995). ISO SPICE technical reports and related artifacts will be field tested. United States involvement in and influence on these international standards should be coordinated and led by the SEI. These include the Baseline Practices Guide (BPG), an ISO product similar to the CMM, as well as appraisal methods, a questionnaire, and process improvement directions based on the BPG.

CMM Maintenance (1995-1996). CMM Version 1.1 requires continuous maintenance, including processing change requests, answering questions from the community, and delivering technical reports as appropriate. Introductory and advanced training are also related to maintaining the CMM.

CMM Version 2.0 (Revision of Version 1.1) (1996-1997). Publication of a revised and updated CMM is planned. Investigations of best practices, level 4 and 5 practices, reflection of ISO SPICE compatibility, and response to change requests will be addressed in this activity. An advanced CMM course will be delivered, and technical reports will be issued as appropriate. Community review of this activity is also anticipated.

3.2.4.7 Related TO&P Activities

No TO&P funds have been identified to support CMM, SECMM, and standardization activities. Potential sources of TO&P include the National Institute of Standards and Technology (NIST).

Partial funding for the PMCMM will be provided via TO&P in 1995.

3.2.5 Process Definition

This section discusses the product activity area related to defining, piloting, and installing processes.

3.2.5.1 Problem Statement

Many software organizations lack the capability to define manageable processes and perform them with fidelity. To be manageable, a software process must be defined and documented, measured, controlled, and continuously improved and optimized. A defined and documented process has its inputs, outputs, work activities, and responsibilities outlined and delimited. Elements such as inputs, outputs, and resources are measured to provide a basis for control and improvement. To control a process, a predetermined mechanism, e.g., a development plan and reporting of status against the plan, must exist to maintain a process in its desired state. Finally, to continuously improve and optimize a process, a predetermined change mechanism, e.g., process monitoring and feedback, must also exist.

In addition to organizational processes, and in support of them, related individual processes must be defined, measured, and followed with fidelity. Quality techniques can be taught and applied at the individual software engineer's and engineering team's level, compatible with the CMM, to enhance transition of organizational processes and disciplined software engineering techniques.

Maintaining fidelity when performing/executing defined processes requires understanding and buy-in to the process and recognition of the value of the process at all levels of the organization. Using a change mechanism, rather than abandoning the process at times of crisis, would greatly enhance current software organizations' processes.

3.2.5.2 Customers

The software engineering community at large, including government agencies, subcontractors, industry, and acquisition agents, are users of improving software processes. The SPI community can leverage the process definition techniques, products, and services to develop or improve software development processes. This effort will transition the CMM into the state of the practice of software development and mature the practice of software engineering in development and supplier communities.

3.2.5.3 Rationale

1. Define, pilot, and install processes.

The current state of the practice is shown in Figure 3-3, where 75% of assessed organizations are at CMM level 1. Where organizations have put in place a sustained, long-term SPI effort, their maturity has improved (Figure 3-4). A key and critical element of the SPI effort (Figure 3-6) is the defining, piloting, and installing of processes in response to findings from appraisals.

2. Transition process definition technology.

There is strong community interest in process-related products and services that the SEI provides to support the definition of manageable processes. This interest is reflected in workshop attendance, requests for on-site engineering services, supplying of resident affiliates available to assist with technology transition, and participation in advisory groups. In addition, needs analyses have identified the need for:

- A definition method
- Definition standards
- Definition guidelines
- A framework for CMM-based processes
- Industrial-strength examples of processes
- Training

3. Integrate process definition and measurement.

Additionally, defined processes must be measurable and measured. Process definition and process measurement are interdependent. Through that integrated relationship they also are able to be analyzed, and the benefits of defined processes being followed with fidelity can be reported. The reporting of the business results of process definition is an ongoing SEI activity (see Figure 3-9).

3.2.5.4 Benefits

Process definition is a prerequisite technology for improving software engineering processes. As indicated previously, processes need to be defined and documented, measured, controlled, and continuously improved and optimized in to be manageable. The definition of processes is the first step in this series of improvement steps. This defining of processes begins the institutionalization of the practices of the CMM into the state of the practice within an organization. Quality process definition is also a prerequisite for attaining CMM level 3 and above.

Process definition provides a basis for communicating about processes. Once processes are defined and documented, they can be reviewed, taught, and understood. Once they are understood throughout an organization, they can be deployed. Based on the results of analyzing and comparing deployed processes, processes can be changed and improved in controlled ways. Finally, defined and controlled processes can be automated to realize the benefits that technology brings to software engineering practice (see Figure 3-13).

3.2.5.5 One-Year Objectives for 1995

Despite progress in software process definition, many organizations are still struggling with the early phases of SPI. Additionally, advanced process definition techniques and state-of-the-art practices are required by higher maturity organizations. A collection of methods, guidelines, criteria, and models needs to continue to be assembled to help organizations implement documented, disciplined processes. These techniques cover the specification, design, tailoring, and implementation of improved software processes. Formal software process maturity models, automated process analysis, and process enactment research will also be conducted. Finally, train-the-trainer support is needed to transition the guidelines to a wider audience.

The software process definition guide is a baseline core area activity. We will complete a process specification standard and process evaluation checklists in 1995. We will refine a prototype process definition method to include process tailoring, planning for process change, and instantiation of defined processes.

Key Items	State of practice as of:			Impact/Metrics
	Today	+2 Years	+4 Years	
Develop and transition process definition method Develop and transition process specification standard	• process development is ad hoc • methods for definition of processes do not support the objectives of process management and the CMM • process content is highly variable and definitions lack key information	• a defined method that supports the CMM and process management is available for trial use • a prototype standard form for process definitions is in trial use • prototype has well-defined syntax and semantics	• method has been tested and accepted • method is readily available • a standard form for process definitions has been tested and accepted • standard is readily available	• processes are engineered using established principles • development and deployment costs are reduced • common form for process definitions facilitates technology development • process definitions can be readily exchanged, facilitating process reuse
Develop and transition software process framework and process references library	• community lacks a framework for defining processes that support the CMM KPAs • process design information in the CMM is high-level • definitions are hard to evaluate against the CMM	• CMM process framework is widely available • prototype process fragments based on the framework are available	• CMM-based process fragments are available • process fragments are expressed in the standard form	• process framework and supporting fragments facilitate process reuse with resulting benefits: leverage, quality, cost
Develop and transition process definition methods, standards, CMM-based process framework, and process fragments	• process development is ad hoc • methods lack support for process management and the CMM • process definition training is not widely available	• process definition training that supports the CMM and process management is available • initial train-the-trainers capability is available	• process definition training is available through licensed vendors, corporate training centers, continuing education providers, National Technological University, etc.	• processes are engineered using established principles • development and deployment costs are reduced

Figure 3-13: Trends in Software Process Definition

Study and transition of personal software process productivity and quality techniques is demonstrating that teaching individuals the techniques and discipline of software process definition and measurement is beneficial to project and organization quality and productivity results. Continued teaching and transition to much larger numbers of practitioners is needed to have the impact that this approach promises.

3.2.5.6 Work Outputs

Software Process Definition Guidelines (1995-1996). Two technical reports are planned for 1995 to document recommended methods and standards for process definition. One will focus on the "data" aspects of a process definition, including standards for information content and a recommended structure for practitioner-oriented process definitions, as well as corresponding evaluation checklists. The second will focus on the "function" aspects of how to develop process definitions, including a prototype method as well as guidance on a variety of method implementation issues arising in practice.

In 1996, experiences with usage of the guidelines documents distributed in 1995 will be gathered. In addition, experiences with higher maturity organizations will be captured and transitioned to those attempting to move up the maturity scale. Correspondingly, revisions to the guidelines documents will be produced as appropriate. Also, examples of leading-edge process definitions will be developed as samples and "how-to's" for later adopters; these are expected to define a few selected KPAs at the higher CMM maturity levels, such as Peer Reviews and Quantitative Process Management. These example process definitions are expected to include process architectures and elements from early adopters and from advanced process research, and to incorporate formal syntax, semantics, and behavioral simulation. The examples will be consistent with the CMM and with the Software Process Definition Guidelines standards.

Personal/Team Software Process (P/TSP) Research (1995). Study and transition of an individual's software process productivity and quality techniques are demonstrating that tracking an individual's process definition and measurement is beneficial to project and organization quality and productivity results. Continued field trials of the P/TSP approach will be conducted to complete the investigation of these methods and to develop approaches for their use in practice. Transition to much larger numbers of practitioners is needed to have the impact that this approach promises. We will complete the development of the Personal/Team Software Process (P/TSP) instructor guide and delivery of a selected number of courses under the baseline core area funds.

Requirements for Advanced Software Process Definition Support (1996). Two primary outputs will be produced. First, a technical report will identify advanced capabilities required to support process definition, process management, continuous process improvement, the CMM, and process reengineering. These requirements will include specific, concrete capabilities needed in process support technology, as well as associated techniques and methods. Second, certain needed technology, techniques, etc., will be explored in the form of prototypes and pilot tests. The output of this exploration work will be in the form of the prototype itself, with results and lessons also being documented in a report and/or scientific publication. A representative example of the sort of capability that would be explored is forecasting the quantitative impact (e.g., impact on cycle time, effort, and defects) of potential process changes before putting them into practice on a real project. This capability would be provided through a method,

specific simulation and analysis techniques, and suitable support technology for implementation.

It is anticipated that results of this task will directly further research and development in this important arena, which is vital to supporting process maturation and integration with emerging technology. Moreover, it is expected that this work will influence researchers, research sponsors, and technology developers to provide technology capabilities required by practitioners to support continuous process improvement and CMM level 5 maturity.

Research Collaboration Agreements (1996). The task will involve collaborative research agreements with other leading individuals and groups in the software process research community. Some potential outputs of such collaborations include:

- Simulation approaches for software processes, considering detailed and project-level impacts.
- Definitions of process variables to be used as drivers in studies of software project cost, cycle time, quality, etc., leading to support of process engineering.
- Work on process changes and using the results of experiences with a changed process for continuous improvement.
- Joint work with researchers and practitioners on the role of feedback in software engineering and in software process engineering.

3.2.5.7 Related TO&P Activities

Client support with various services and other government agencies will complement the development of the core-funded products. The clients anticipated in 1995 include the Defense Mapping Agency (DMA) and the Naval Oceanic Office (NAVOCEANO).

The TO&P activities involve training, tailoring the products to specific customer needs, support during initial process definition efforts, and the transition of capabilities to the customer.

3.2.6 Process Measurement

This section discusses the product activity area related to process measurement, empirical methods, and CBA.

3.2.6.1 Problem Statement

Many software development organizations do not use quantitative methods effectively in managing software projects. Many need common or consistent definitions of data to be collected. Some do not collect data or do not collect them in consistent ways, including storing them in accessible databases. Some collect data but need to know how to use them in managing software projects. Some do not know how to analyze the data they collect. They need to gain knowledge or insight from the data to make them useful as predictors of results.

Many software development organizations do not use quantitative methods effectively in measuring improvements in software products and processes. The needs here are similar to those

given above. In addition, historical results in the data need to be maintained and used in planning or evaluating changes made for improvement purposes.

Many organizations do not formally appraise the current state of their software development processes. They recognize that they have cost, schedule, productivity, and quality concerns, but they often apply short-term fixes to address the symptoms of these problems. Appraisals of their own or their suppliers' processes against the CMM have only recently begun to be practiced, and only within a small portion of the software development community. Reappraisals following improvement efforts are only now beginning to be reported. Data gathered to reflect the results of the improvement efforts are not consistently defined, collected, interpreted, or reported.

The software engineering community needs data about the state of the practice of software development to determine:

- Where organizations are relative to the state of the practice.
- Which innovations in the practice contribute to improved performance and capability.
- The value of and return on SPI investments.

3.2.6.2 Customers

Customers include the software development community, for example, developers of software, acquirers of software, and suppliers of appraisals in government and industry.

SPI champions, sponsors, and agents for change are also customers. However, since it is not process improvement for process improvement's sake, the business results of appraising, measuring, defining, and evaluating the results of processes and the changes to processes are meaningful to managers and other stakeholders of an organization as well. In addition, decision makers who must allocate scarce resources to SPI efforts need information on value and progress of SPI.

3.2.6.3 Rationale

Although process measurement is common and seen as integral to process and product control and improvement in other industries, it is not as widely practiced in the software industry. The need for data definition, collection, analysis, and use of the results is recognized as crucial in software development, and many organizations attempt to install a measurement program. However, many organizations want and need assistance in getting started; in understanding measurement techniques and the meaning of quantitative analysis; in communicating about and believing in measured results; and in using measures to estimate, plan, and control projects results (see Figure 3-14).

Also, the software community wants to improve its quality, costs, productivity, and schedule performance. To do this, it has contributed to the CMM and wants to transition the state of the practice toward the vision of software engineering excellence that the CMM elaborates. To such an end, the organizations in the community have stated the need for appraisals that re-

late their or their suppliers' current state and improvement plans to the CMM. Moreover, they have asked for repeatability of appraisal results over time and across different appraisal teams and for consistency across different appraisal types. The SEI process focus area is viewed as an impartial source for satisfying these needs. In particular, we will supply consistent training, authorization programs, and quality control for the community.

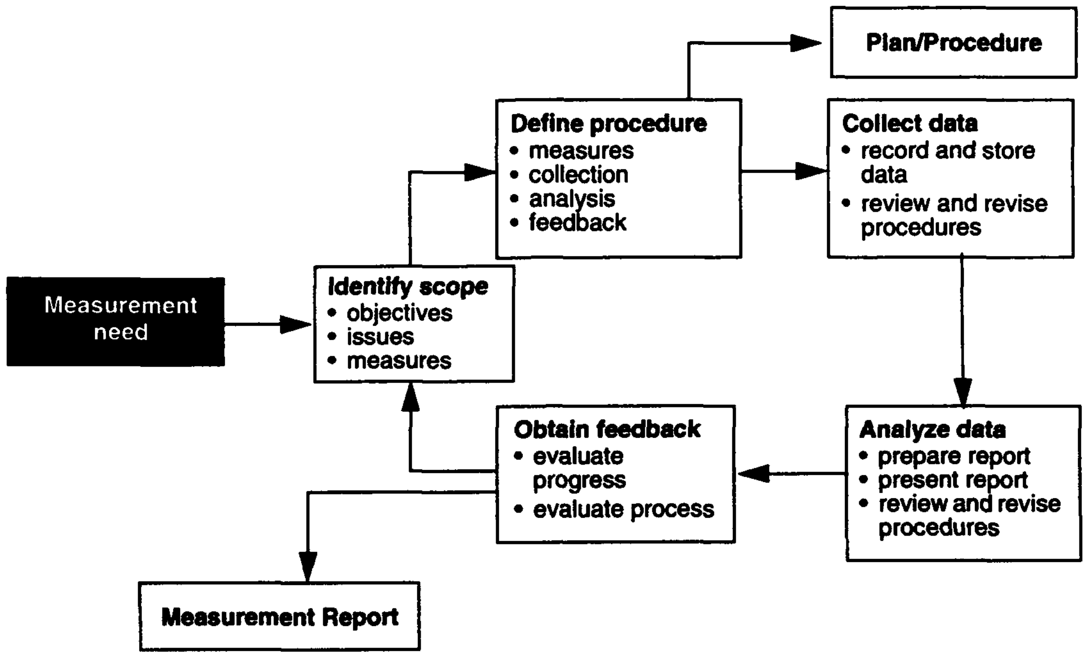


Figure 3-14: Software Process Measurement Concepts

Finally, sponsors and customers are demanding information that validates the value of SEI products and services. Customers are demanding information to substantiate and justify the value of SPI programs. The software community needs a "safe haven" to serve as the custodian for its data and looks to the SEI to be that haven. Successful implementation will encourage the broad masses to participate, will provide a baseline of acceptable performance, and a yardstick for improvement efforts.

3.2.6.4 Benefits

The application of measurement processes, practices, and methods is needed to support advancement in process capability. The SEI brings a measurement focus to the CMM. There is a recognition of the need to begin a measurement strategy at levels 1 and 2, not to wait until level 4, and to build on that strategy as an organization matures over time.

Measurement is an indicator of progress in process improvement programs, and it can be an indicator of bottlenecks and inhibitors to progress as well. Measurement and feedback on the

software process also provide significant potential for insight into areas needing improvement. A mature software process is managed, defined, controlled, measured, and focused on orderly improvement.

All our efforts are focused on improving the ability of organizations to meet their business objectives. Assessing an organization's current process establishes a baseline measure of the process and findings on which to pursue business-related improvements. Evaluating potential subcontractors' processes improves the process of software acquisition and the expected results from the selected contractor. Performing subsequent reassessments and reevaluations provides results for comparison to the baseline, hopefully reflecting measurable improvement. These improvements are aligned with improving business results for both the developer and the customer (see Figure 3-15).

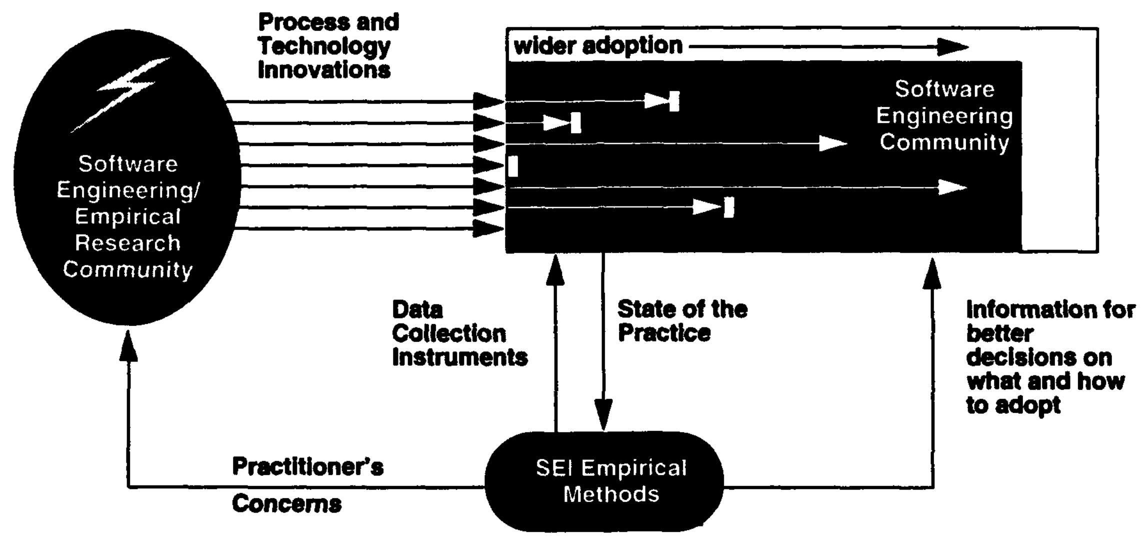


Figure 3-15: Empirical Methods Concepts

Better data on the effects of SPI will guide customers' decisions in investing in software engineering improvements. Gathering, analyzing, and reporting using better data will demonstrate the value of the CMM and its improvement methods. Monitoring improvement in the community will help sustain the commitment to software engineering improvement and continue to build and support the infrastructure necessary to sustain and broaden improvement efforts.

3.2.6.5 One-Year Objectives for 1995

The SEI core measures are serving as the basis for Air Force and DoD policy formulation. The measurement definition work has included working with the broad software engineering community to achieve consensus for an initial set of measurement definition checklists and frameworks. The initial focus was on project management measures of size, effort, schedule, and quality. These measures served as a starting point since they are crucial to managing project

commitments and can serve as a foundation for achieving higher levels of process maturity, as defined in the CMM.

1. Transition software measurement capabilities.

To maintain leadership, additional measures and measurement definitions need to be developed that address level 3 KPAs and beyond as described in the CMM. For example, measures at level 3 become increasingly directed toward measuring the intermediate and final products produced during development, e.g., detailed defect requirements and reuse measures. Measures at level 4 capture characteristics of the development process itself to allow control of the individual activities of the process. At level 5, processes are mature enough and managed carefully enough to permit measurement to provide feedback for dynamically changing processes across multiple projects.

2. Provide and transition products to assess the state of the practice.

One of the most significant and useful SEI products is appraisal methods, in particular, assessments and evaluations. The assessment and capability evaluation methods, which were upgraded in 1994 to CMM Version 1.1 and to a common rating approach, need to be documented for public understanding and in support of transition to partners and collaborators. Additionally, results of field trials and upgrades for higher maturity level organizations, if required, need to be incorporated and/or developed for these and other appraisal methods. Continued integration, maintenance, evolution, and quality control over these methods is required to insure their sustained credibility and value. Findings from appraisals, including software process assessments, IPI, and SCE, will be provided to the SEI by lead assessors and lead evaluators in detailed format that will allow analysis by KPA and practices at each level of the CMM. Receipt of input and verifications back through the appraiser program is required to make data available for use by the SEI and the software community. The upgraded assessment and capability evaluation methods hare common elements. There is a need to make visible the architecture of those common elements, and they need to be documented in a CRF. This will allow reuse of those common elements in appraisal methods developed by the SEI, such as mini-assessment, or in the systems engineering area, or in methods developed by others. The effort is targeted to be funded through baseline core.

3.2.6.6 Work Outputs

This section describes approved core work outputs for empirical methods studies and CBAs. No process measurement activities were included in core efforts for 1995. Process measurement activities are listed in the proposed core add-on activities (see Section 3.2.7.3).

CMM Validity Studies (1995). The goal of this task is to gather data relevant to evaluating the accuracy and usefulness of the CMM as a basis for SPI efforts. A survey conducted at the 1993 SEPG National Meeting indicated that data showing the results of CMM-based SPI was the number one need of SEI customers. We are responding to this need with a set of studies designed to answer the questions of greatest concern to customers, such as the time and cost

of moving up the maturity scale, expected benefits from higher maturity levels, contribution of KPAs to project performance, typical barriers to CMM-based SPI, successful techniques for overcoming these barriers, and contributions of non-CMM factors to project performance. This ongoing effort will result in a series of conference presentations, publications, and technical reports. In 1995, we will conduct a survey of software organizations that have been assessed to gather data on their SPI efforts and will report those results to the community.

Our experience in 1994 has shown that many organizations are not gathering data on their process improvement efforts. In 1995 we will form a working group to develop and pilot a methodology for measuring the results of their process improvement efforts. These methods will subsequently be used to transition this capability into software organizations. By doing this, we will be seeding the sources of data needed to address the issues associated with validation of the CMM (cited above).

SEI Process Database (1995). The SEI process database is a repository for information from software process appraisals. This ongoing activity provides support and infrastructure for many of the planned empirical studies of the software community. The basic scope of the database provides the capability for the production of reports on the state of software process maturity and current process-related problems among software organizations. Over the past two years, the volume of data reported to the database has exceeded the volume of all previous years combined and is expected to continue to grow. To keep up with the flow of data, the development of the database itself has been progressing along a plan that focuses on efficiency of data entry and reporting, integrity of data, expandability of scope, and maintainability. In 1995, the database will continue to develop along the lines described. Data from the CMM validity studies and value of the SEI studies will be integrated and stored in the database. Having all of these data in a single relational database will permit richer analyses of the software community. Also in 1995, we will continue to produce updates on the maturity profile of the software organizations and will report on the software process issues confronting these organizations.

Appraisal Methods (1995). Two of the key products of the SEI are its well-accepted appraisal methods. We have an obligation to further these techniques and to transition them throughout the software community. The assessment and capability evaluation methods, which were upgraded in 1994 to CMM V1.1 and to a more common rating approach, need to be documented in technical reports for public understanding and in support of transition to partners and collaborators. Additionally, results of field trials and upgrades for higher maturity level organizations, if required, need to be incorporated and/or developed for these and other appraisal methods.

Value of the SEI (1995-1996). As requested by the SEI JAC, this effort will expand the scope of the work done under the CMM validation study and the SEI process database effort to include the people and technology products and services of the SEI. It will focus specifically on the value accrued by software organizations using SEI products and services for addressing

technology and people-related issues. This ongoing activity will result in a series of technical reports and white papers for use within the SEI. A plan, data collection instruments and methodology, and initial data collection will occur in 1995. This work will require the development of an SEI-wide customer database and close collaboration across the technical focus areas and divisions of the SEI.

Appraisal Architecture and CRF Report (1995). The upgraded assessment and capability evaluation methods share common elements. There is a need to make visible the architecture of these common elements, and these need to be documented in a technical report on the CRF. This will allow reuse of these common elements in appraisal methods developed by the SEI, such as mini-assessments, or in the systems engineering area, or in methods developed by others. Such an architecture needs to be available so that various appraisal methods can be compared and appraisal results can be consistent.

3.2.6.7 Related TO&P Activities

Client support with various services and other government agencies will complement the development of the core-funded products. Clients anticipated in 1995 include the DMA, the AMC, and the Defense Information Systems Agency (DISA).

The TO&P activities involve training, tailoring the appraisal and measurement products to specific customer needs, support during initial process measurement efforts, and transitioning capabilities to the customer.

3.2.7 Proposed Add-On Activities

The following section describes proposed core add-on activities in the process focus area. The SEI is asking selected members of the software community to evaluate the relative attractiveness of these add-on proposals as possible elements of the final 1995 core program, as funding permits. The proposals are grouped according to the product activity areas within this focus area. Appendix A lists all baseline and add-on items.

3.2.7.1 Process Maturity Modeling

SP-1A Systems Engineering CMM (SECMM) Development and Integration

The objectives for the SECMM are included in the add-on core area. The results of the 1994 efforts related to the SECMM will include the model description and practices, appraisal method description, and pilot appraisal report. To establish the model firmly as a driving force within the systems engineering community, and to ensure that the efforts of the systems engineering community flow to the software engineering community, will require the development and deployment of additional products. Some products are appropriately funded via a continuing collaboration with SECMM participants; others are more appropriate for SEI funding. The latter include:

- Technical report defining methods for integrated (both systems and software) engineering appraisals.

- Technical report defining methods for integrating systems engineering process improvement into existing SPI programs.
- SECMM seminar materials to provide education related to SECMM to both the systems engineering and software engineering communities.

In relation to integration, significant discussion has begun within the software community, which points to the need for guidance in effectively integrating systems engineering improvement into the organization in a systematic way that does not disrupt ongoing SPI efforts. The focus of the above efforts is to smooth the transition toward integrated approaches in environments where both SWCMM-based and SECMM-based improvement are being pursued.

SP-2A People Management Capability Maturity Model (PMCMM)

The PMCMM is targeted to be partially funded by add-on core. It has been shown in several studies that human talent applied to software development is the strongest predictor of its outcomes. Further, software development of sizeable systems must transition from a mystique of individually creative artists to a team-based profession that emphasizes continuous learning. Thus, improvement in the capabilities of the professionals developing software must go hand in hand with process improvement. Many software organizations have wanted to improve their ability to motivate and develop their technical staff as part of their improvement program. Since the CMM offers little guidance on these issues, they are having difficulty deciding how to systematically increase the talent of their organization. Rather than just focusing on individual human resource practices such as training, the PMCMM will focus on maturing an organization's ability to develop talent through a whole range of practices. This can be integrated with the CMM and with ongoing work in an organization in SPI.

The work for which add-on core funds are sought involves completing the development of the PMCMM during 1995 and releasing a first version nationally for use with improvement programs. Version 1.0 of the PMCMM will be developed through a process of national input and review in much the same manner that the CMM was developed and reviewed. A senior advisory board has already been established and has made inputs into a skeleton draft of the PMCMM. A national review and correspondence group will be established and coordinated. A national workshop will be conducted to attract broad input into the formulation of the model. Version 1.0 is targeted for release in mid-1995. Add-on core funds will also be used to develop a method for assessing an organization against the PMCMM. This method will be consistent with and complementary to process assessment methods already in use. Application of the PMCMM to improvement programs will be performed with TO&P funds that have already been committed.

SP-3A Systems Engineering Capability Maturity Model (SECMM) Maintenance

The maintenance of the SECMM is targeted to be partially funded by add-on core. Although the industrial community participating in generating the 1994 work products is interested and willing to provide some level of continuing funding, it is not anticipated that sufficient funding from industry sources will be available to maintain the model during its early, critical-growth stages. The proposal for core funding would cover SEI resource needs related to community involve-

ment, planning of future revisions under the direction of the Steering Group, funding the interaction of the SEI as a focal point for receipt of change requests to Version 1.0 of the SECMM, working with the Steering Group in the disposition of received change requests, and coordinating the revision work.

3.2.7.2 Process Definition

SP-4A Process Research Program

Support for integration of Personal/Team Software Process (P/TSP) offerings and transition of them into organizations, including pilot testing at selected client sites, is proposed to be funded through add-on core area funds. Additionally, development of representative process fragments for P/TSP, developing and conducting automation enactment trials, and planning and transitioning a licensing program for P/TSP is proposed to be funded with add-on core.

SP-5A Software Process Definition Framework

Development of the software process definition framework is proposed as an add-on core activity. The framework is a technical report providing a reorganization of the CMM information, in a form designed to further facilitate the task of defining processes that are consistent with the CMM. In 1995 we will initiate development of corresponding process fragments that elaborate and instantiate the basic framework. These prototype fragments will be structured by CMM KPA and documented in report form, will be refined to lower levels of detail than the current CMM content, and will serve as a framework (i.e., structural basis) for organizations to directly use in developing fully detailed organizational process definitions. These fragments and the basic framework will be developed to be consistent with the emerging software process definition guidelines, and will eventually support CMM Version 2.0. By easing the work of defining processes consistent with the CMM, this collection of fragments is expected to accelerate process maturation.

The project's training offering must also be upgraded to reflect this work and properly integrate the material. Train-the-trainer support is also needed to transition the impact of the framework and fragments to wider audiences. These training enhancements will also be part of this add-on core-funded effort.

3.2.7.3 Process Measurement

SP-6A Process Value Method

The data supporting state-of-the-practice analysis and reporting will also support a process value method study, which will produce a method for project managers or SEPGs to provide a perceived value, from a business perspective, from a proposed process change. The change can be at the level of an appraisal, a KPA, or a particular activity within a KPA. The method would allow the organization to understand what would be required in resources to make the process change, e.g., introduce inspections or peer reviews, as well as the expected results or value of the change. The method, to be documented in a technical report, would support both quantitative and qualitative aspects to determine value. Add-on core is targeted for this proposed effort.

SP-7A Software Measurement Handbook

A software measurement handbook needs to be developed to serve as a single source for how software measurement programs can be initiated, sustained, and broadly institutionalized. The handbook would be a guide for industry and government managers and practitioners and would include:

- Reasons why measurement is important/benefits
- Examples of the use of software measures
- Infrastructure issues
- State of the practice
- A software measurement hierarchy
- Measurement definition frameworks for the SEI "core measures"
- Role of measurement in the context of the CMM
- "How to" design and implement an effective software measurement process
- A software measurement bibliography
- "How to" start a program
- "How to" sustain a program
- Case studies

The effort to produce such a handbook, recommended by the SEI Measurement Steering Committee as well as our customer community as a high priority work product to assemble in 1995, is proposed to be targeted in the add-on core activity area.

SP-8A Software Product Measurement Definitions

The SEI core measures are serving as the basis for Air Force and DoD policy formulation. The measurement definition work has included working with the broad software engineering community to achieve consensus for an initial set of measurement definition checklists and frameworks. The initial focus was on project management measures of size, effort, schedule, and quality. These measures served as a starting point as they are crucial to managing project commitments and can serve as a foundation for achieving higher levels of process maturity, as defined in the CMM.

However, to maintain the momentum of enhanced measurement programs in the community, additional measures and measurement definitions need to be developed that address CMM level 3 KPAs and beyond. For example, measures at level 3 become increasingly directed toward measuring the intermediate and final products produced during development. Measures at level 4 capture characteristics of the development process itself to allow control of the individual activities of the process. At level 5, processes are mature enough and managed carefully enough to permit measurement to provide feedback for dynamically changing processes across multiple projects.

Add-on core funding is needed to expand the initial software measurement definition effort and upgrade the software measurement course to include software product measurement. Add-on core is targeted toward expanding the initial software measurement definition effort. A next

logical follow-on effort would include undertaking measurement definition activities pertaining to quantifying and defining software product characteristics, e.g., detailed defect measures, product reliability measures, and complexity measures.

SP-9A Empirical Methods Product Improvement

The process focus area has developed a number of instruments used in other process area products that collect data from the software community. These include the Software Process Maturity Questionnaire and the Organization and Project Questionnaires, both of which are used in software process appraisal methods (CBA-IPI, SCE, and Interim Profile).

The appraisal methods assist in the improvement of software organizations and also serve as an important source of data for characterizing the state of the practice. These questionnaires play a dual role of identifying the requirements of an appraisal and providing data for SEI ongoing research and development efforts. Improvement of these questionnaires is vital both to the appraisal methods and to the SEI efforts that rely on the data that these questionnaires capture.

To enhance an ongoing improvement effort, change request and other feedback mechanisms have been developed and deployed for these questionnaires. Funding for this effort is needed to analyze the change requests and feedback received by the end of the second quarter of 1995 and to update these questionnaires accordingly.

3.3 Risk Management

As systems become more complex, the risks associated with their development increase. At the lower level of complexity, engineers and managers can assess and manage the risks based on personal knowledge and experience. But as complexity increases, individual knowledge and experience soon become inadequate. Engineers begin making inappropriate technical decisions, and managers are routinely surprised by cost overruns and schedule delays.

What often happens next is seeking outside expertise to gain more specific knowledge and experience. However, at some point the system becomes so complex that the interaction and communication among the managers, engineers, and experts must be facilitated and managed consistently. At this point, a systematic and structured set of processes, methods, and tools for assessing and managing risks becomes imperative (see Figure 3-16).

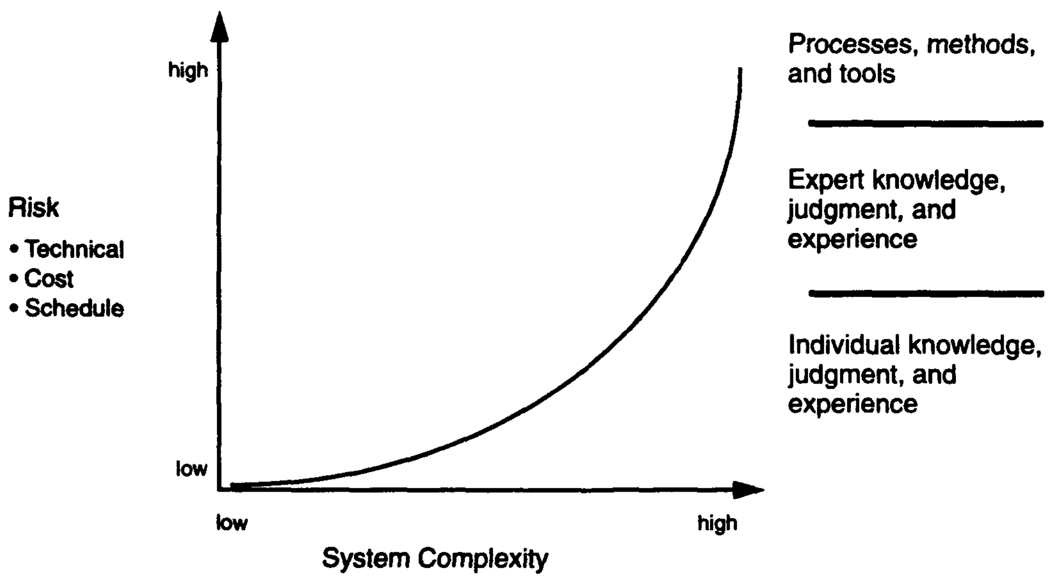


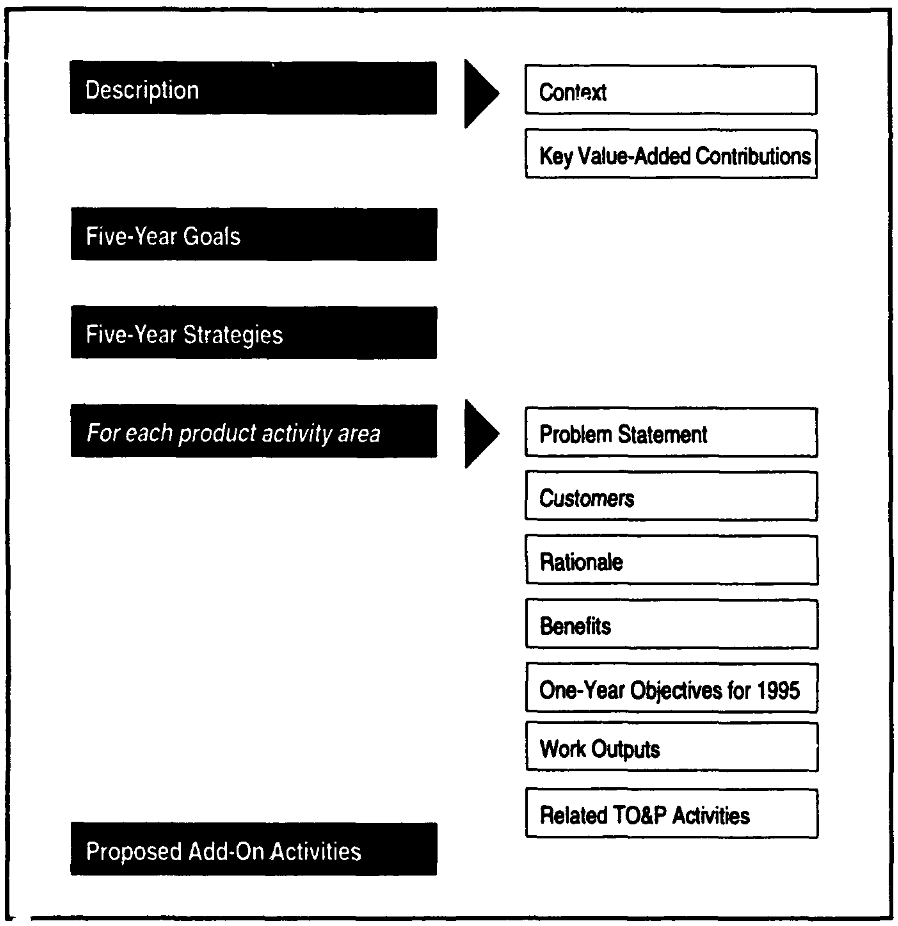
Figure 3-16: The Need to Manage Risk Increases with System Complexity

Without a structured framework, people have difficulty working effectively as a team to solve complex problems. The lack of such a framework accounts for some of the conflict seen in the integration of products, methods, and tools in system development today. Area specialists understand the need for their particular discipline in the overall system development process but not necessarily how it affects other disciplines. There are numerous examples, including:

- Cost and schedule decisions on technical performance.
- Requirements changes on existing architectures.
- Vendor changes in computer-aided software engineering (CASE) tools on the system design process.
- Process changes on product development schedules.

Proactively addressing uncertainty is a rational approach in developing and implementing sound program strategies—strategies that weigh program opportunities and risks from both business and technical perspectives. For example, one would develop a program strategy for reengineering a product by weighing the advantages of the technology and processes with the uncertainties perceived from the existing and predicted conditions. A risk-aware approach supports any environment where change or uncertainty are recognized factors, one that leverages opportunities and deals with the risks. Reengineering, reuse, large-scale development, and unprecedented technology are a few examples where risk management would be effective in the acquisition and development processes as well as in the program's product. By focusing on risk management, programs can identify and analyze software technical uncertainty to present the decision maker with the right information in a timely fashion. Since such generally accepted methods do not currently exist for software, the SEI is developing processes and supporting methods that will systematically identify and analyze software technical uncertainty. We seek to improve system acquisition and development by identifying key risk management practices and the criteria by which they can be integrated into software development processes and program management.

The sections for this focus area are:



3.3.1 Description

3.3.1.1 Context

Risk management contributes to maturing both process and technology in the SEI strategic framework described earlier in this chapter. Risk management contributes to maturing the process by defining and modeling processes for assessing and managing risks in software-intensive systems. These processes are described in a risk management paradigm (see Figure 3-17).

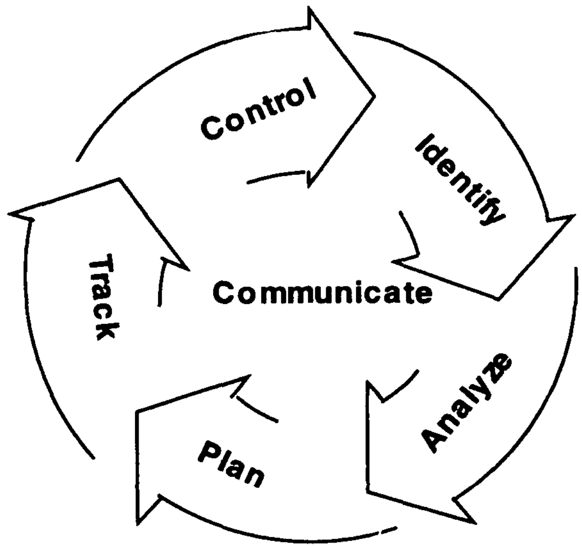


Figure 3-17: Risk Management Paradigm

During 1994, the emphasis on risk identification shifted to develop other methods in the paradigm and to field test risk management methods with customers in industry and government. The field testing has demonstrated the dual use of these methods in government acquisitions and commercial systems.

Risk management contributes to maturing the technology by identifying the technical issues and needs associated with developing software-intensive systems. One risk assessment method, the Taxonomy-Based Questionnaire, has gathered data on over 30 projects in both the real-time and information system domains, identifying issues in the software engineering process, methods, and tools.

Work continues on the Questionnaire for the identification of development risks. The Questionnaire evolved through a research program drawing on software development expertise, the analysis of software development literature, and analysis of data collected through applying the Questionnaire in a series of field tests on software-intensive projects within various gov-

ernment and civilian organizations. It has proved to be an effective method not only for identifying project risks, but also as a catalyst stimulating communication within projects.

The Questionnaire is based on a taxonomy of software development for identifying risk throughout the development life cycle. The software taxonomy is organized into three major classes:

- **Product Engineering**. The technical aspects of the work to be accomplished in producing the product.
- **Development Environment**. The process, methods, and tools used to produce the product.
- **Program Constraints**. The contractual, organizational, and operational factors within which the software is developed but that are generally outside the direct control of local management.

The aggregate of this data indicates that the risks for software-intensive systems are fairly well-balanced across the three classes; however, any specific project may have significantly more risks in any one of the three classes (see Figure 3-18).

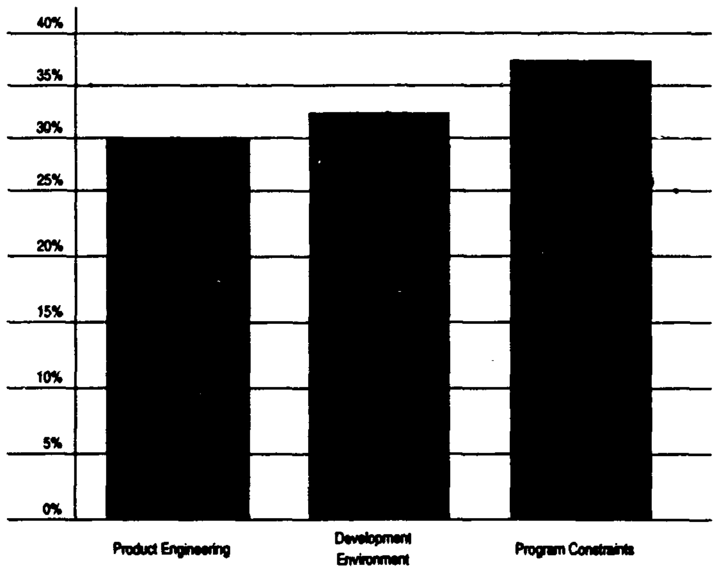


Figure 3-18: Risk by Class

Risk management methods are also applicable in other SEI work. Risk management will become a key practice in the CMM, and is already used as a KPA in the SEI SECMM.

Software risk evaluations (SREs) will become a general assessment process to assess client needs for software engineering improvement activities. The SRE teams, composed of people with software engineering expertise and people from client organizations, are designed to maximize the knowledge in risk assessment, technology, and the project domain. Participation

by the SEI enables the SEI to build and maintain a core competency in risk assessment and allows participants to gain knowledge and insight into the state of the practice of software engineering.

In 1994, the SEI developed a three-day risk identification training module. This module trains people in how to use the Questionnaire and the probing protocol used to elicit data during the interview sessions. A one-day risk management tutorial was also developed and presented in 1994. This tutorial is specifically designed to raise organizational awareness of the value of risk management in increasing the probability of project success. The tutorial provides insight to executives and middle management about software development risk and what it means to the success or failure of a software-intensive project. The tutorial provides the foundation for additional details of the planning and analysis training course module being developed in 1995.

The Navy Program Executive Officer (PEO(A)) continues to provide the SEI with unique opportunities to evaluate risk management approaches in both government and industry to develop and test its team risk management concept in active programs. As a result, the methodology is being used to manage risk in two active programs. From this initial and ongoing use of team risk management, a user's guide and tutorial have been prepared for pilot use. The research and field implementation activity enabled the SEI to build and maintain a core competency in risk management. The ongoing collaboration with government and industry provides the opportunity to develop and test other methods and concepts such as risk analysis, action planning, and risk metrics.

3.3.1.2 Key Value-Added Contributions

Risk management processes, methods, and tools provide the conceptual foundation and building blocks to establish and institutionalize risk management in the acquisition and development of software-intensive systems. The following are the major thrusts:

- Developing standards in risk management.
- Establishing processes, methods, and tools to manage risk in acquisition and development.
- Developing training in risk management.
- Establishing customer-supplier team risk management.
- Developing methods to identify risks associated with introducing technology into program products and processes.
- Collecting, analyzing, and disseminating empirical data and information on risks and their corresponding mitigation strategies.
- Developing qualitative and quantitative methods to enhance risk analysis and decision making.

Developing models and tools to aid risk management and decision making. Based on the data from the 30 risk assessments described in Section 3.2.1.1, we found that programs need a

structure to assess software risks, to gather data about risks, and to share information with other projects regarding issues, mitigation strategies, or lessons learned regarding software.[1] Unfortunately, numerous textbooks on risk management offer little on how to implement a risk assessment or management process. The SEI is chartered to gather this type of information and share it with the community. The SEI is evolving several product lines to enable the community to make better decisions regarding risk management.

Use of the Taxonomy-Based Questionnaire is currently being transitioned to the community through technical collaboration agreements (TCAs), TO&P customers, conferences, tutorials, and training courses. The Questionnaire method has been incorporated into the SRE process and is the foundation for incorporating risk management into the source selection process. The Questionnaire provides a structure for the SEI to gather and report data across the entire acquisition life cycle for programs in different services and domains. The use of the Questionnaire as a common assessment mechanism enables the SEI and the software engineering community to share assessment data and map specific risk assessment information on mitigation strategies through a risk data repository (described later).

Team risk management is a new paradigm for managing programs or projects by developing a shared product vision, focused on results, and using the principles and tools of risk management to cooperatively manage risks and opportunities. Team risk management uses a framework of building teams with both customer and supplier (e.g., government and industry) to systematically and proactively manage risk. The SEI is well suited for this effort since it is regarded as a neutral party by both government and industry.

3.3.2 Five-Year Goals

To raise the maturity of the practice and management of software engineering, the SEI is developing and transitioning products and services enabling software engineers and managers to better identify and manage technical uncertainties in software-intensive systems. This improvement will be achieved by providing proven and tested methods and processes in the form of education, training, questionnaires, and data gathered from government and civilian organizations.

There are three primary goals:

- Improve risk management in acquisition and development of software-intensive systems.
- Establish a national repository of common risks and their mitigation strategies.
- Establish software risk identification and action planning as a foundation for software engineering improvement.

1. Some organizations such as consulting firms do possess a risk assessment and management process but treat those processes as proprietary and keep their data confidential.

Very little data exist today on which to base quantitative measures of success in risk identification and management efforts. By 1999 we will have in place a system allowing quantification and tracking of method effectiveness in reducing risk, and evidence that our techniques are effective. In the meantime, we are using the following as measures of success:

- Evaluations provided annually by our TO&P customers via "sponsor feedback forms."
- Number of clients who collaborate with us in independent risk assessments and team risk management activities.
- Number of organizations adopting the SEI risk management process.
- Number of attendees at risk identification training courses.

We expect that by 1999, programs will be using validated risk management methods systematically in acquisition and development throughout the life cycle. Also, a new paradigm of team risk management will bring the customer and developer together in a cooperative spirit of proactive risk management. In addition, we expect to achieve the goals shown in Figure 3-19 by 1999.

Additional 1999 Goals	Measures
Software risk management will be recognized as a key practice in the acquisition life cycle.	• Risk management will be established in government and civilian standards with documented methods and processes. • System acquisition strategies and decisions will be based on information from proactive, systematic, and validated risk management methods. • Formal program reviews will be conducted using systematic and validated risk evaluation methods. • The DSMC will have courses on software risk management in acquisition.
Software risk management will be recognized as a key practice in the development of software-intensive systems.	• Major DoD contractors will have incorporated risk management into their software development process. • Risk management will be established as an organizational capability, executed on a continuous basis, and integrated within the context of program management. • The SEI will have an established national repository of risks and supporting risk reduction strategies based on information gathered from strategic partners and user networks. • Commercial training organizations will have continuing education offerings on software risk management.
SEI team risk management will be adopted as the benchmark for integrated product teams using systematic and validated methods and processes.	• Major government programs will have integrated customer and developer risk management activities—team risk management. • Team risk management will have integrated technical, cost, and schedule risk management into continuous, routine program management in major programs. • The DoD will have adopted team risk management as its acquisition and development practice.

Figure 3-19: Additional 1999 Goals

3.3.3 Five-Year Strategies

The SEI five-year strategy for risk management has three primary components:

- Develop and test risk management processes, methods, and tools and incorporate them into the acquisition and development process.
- Establish the Taxonomy-Based Questionnaire as a common method for assessing risks and the impact of new and existing technology on software-intensive systems. This includes using the Questionnaire as one of the foundation assessment methods for software engineering improvement.
- Catalyze and motivate the acquisition and development community to adopt risk management processes and methods through community interaction (workshops, tutorials, training, TO&P customers, strategic partners, and conferences).

To be effective, we must not only provide methods for identifying and managing technical uncertainty, but we must also provide methods to facilitate the communication of risk issues. In addition to the usual cultural barriers, the common negative perception of risk makes change even more difficult. The communication process must de-personalize risks so they are viewed as opportunities for program success. All the method development and field testing activities directly address communication to enhance or enable effective communication.

In the acquisition and development areas, products will establish capabilities to evaluate software technical risks for programs, either by a customer or independent agent. The potential products cover risk evaluation and independent risk assessment methods and their application within the acquisition community. The Army (CECOM SED) plans to use the risk taxonomy in a source selection evaluation process to strengthen the acquisition process.

Team risk management promotes a new paradigm of shared commitment to manage program risks as a team of customer and supplier (e.g., government and industry). Team risk management creates an environment built upon a set of systematic processes, methods, and tools that enable the customer and supplier (developer) to work together to manage risks throughout the life cycle. It is built on a foundation of risk management and cooperative team principles. This effort is being pilot tested with the Navy (PEO(A)).

The Questionnaire is being established as a common method for assessing risks and the impact of technology on software-intensive systems. The method provides a foundation for gathering data and structuring a data repository. The taxonomy classes and elements will be the basis for categorizing information in the repository.

Software engineering consists of process, technology, and people. The SEI risk taxonomy provides a method for assessing needs and risks in these areas for an acquisition and development effort. The identification of these needs will provide a basis for developing an improvement strategy and priority for implementation. The Questionnaire method will also enable the client to develop both short- and long-term improvement strategies.

To ensure a successful transition strategy, we are approaching transition systematically by targeting products to leverage limited SEI resources and to support adopting and sustaining the technologies. This includes community awareness activities such as conducting the annual SEI Software Risk Conference, presenting risk management tutorials at conferences, SEPGs, and customer and developer working groups. These activities provide the additional benefit of feedback on our work. Education, training, collaboration, and publications will be our primary instruments for affecting understanding. For example, education and training include the risk identification training course and software risk management tutorials. Collaboration is addressed by the field activities to test specific methods with our government and industry partnerships. Installation is addressed by teaching leadership courses on risk management and conducting independent risk assessments. Finally, we are addressing adoption and institutionalization by conducting and transferring team risk management, risk management improvement, and independent risk assessment activities.

We will use an advisory board to provide advocacy and community feedback on strategy and outputs. The risk advisory board will be comprised of respected members of the software engineering community who are knowledgeable about risk management. The members will be drawn from industry, government, and academia, as follows:

- Industry: Drawn from different corporations within the defense and aerospace sector or from different corporations within the commercial sector.
- Government: Drawn from different branches of the armed services and different agencies.
- Academia: Drawn from different universities. Having one member from CMU will be encouraged.

3.3.4 Risk Management in Systems Acquisition and Development

This section discusses the product activity area related to risk management in systems acquisition and development. In 1995, we are placing emphasis on systems acquisition by broadening the scope and application of our methods and proposing work on a maturity model for systems acquisition. With our continuing success we are broadening the scope and distribution of risk management processes and methods.

3.3.4.1 Problem Statement

Acquisition and development programs continue to suffer large cost overruns, schedule delays, and poor technical performance. This is a general result of failing to appropriately deal with uncertainty in the acquisition and development of complex, software-intensive systems. The acquisition and development community, both government and industry, lack a systematic way to identify, communicate, and resolve technical uncertainty. Often the focus is on the symptoms of cost overruns and schedule delays rather than on product acquisition and development root causes. In fact, all areas in systems development are potential sources of software risks (see Figure 3-20).

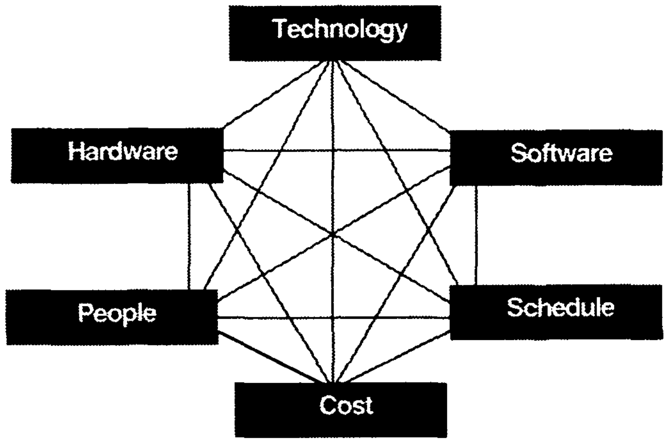


Figure 3-20: Risks Within a System Context

3.3.4.2 Customers

The potential customer base is nationwide and spans defense, non-defense, government, and commercial industry. Specifically clients include:

- Navy Program Executive Office for Air Anti-Submarine Warfare, Assault and Special Missions
- Navy (NAVOCEANO)
- Marines (MCTSSA)
- Army (CECOM/SED)
- U.S. Treasury Department
- Loral, Boulder, Colo.
- Northrop Corporation
- Harris Government Information Systems Corporation
- Chrysler Technology Airborne Systems Inc

3.3.4.3 Rationale

Every organization involved in developing software-dependent systems has the problem of being able to identify risks early enough to take corrective action and avoid surprises that could jeopardize the success of their programs. These organizations also need to be able to assess and plan the technology improvements to build tomorrow's complex systems.

In a survey conducted at the 3rd Annual Risk Conference in 1994, only 10% of the respondents reported that they do risk management on a regular basis. Since 1991 we have conducted risk assessments and participated in developing risk mitigation strategies. The data from this field work show the need for a cost-effective and practical means to identify, analyze, track, and mitigate risks. However, feedback from SEI workshops and this field work has re-

vealed that risk management processes, methods, and tools are simply not widely known or used. Teams with multiple and varied disciplines require processes and methods to identify, manage, and communicate the risks in the acquisition and development process.

The government consistently asks for guidance in risk management for developing acquisition strategies, assisting in source selections, and managing a contract life cycle. We have an ongoing effort with both the Army and Navy in which they have requested direct support to put risk management explicitly into the source selection process.

The Risk Taxonomy-Based Questionnaire is increasingly being adopted by industry and government as an integral part of their acquisition and development processes for risk identification. Industry in particular is requesting the Questionnaire be expanded to cover the entire system life cycle from concept through post-deployment software support (PDSS). Specific areas of interest are performance, maintainability, security, and post-deployment support. Furthermore, field test results have shown a lack of consistent methodology for identifying and managing risks.

As systems become more complex, a continuing improvement initiative is needed to stay abreast of technology to increase efficiency and to take advantage of the latest techniques to produce today's complex systems. There are no roadmaps, however, for organizations to follow in efficiently improving their technology base to ensure that they build the best quality products at the lowest cost with the least amount of risk. While an SCE provides insight into contractor management processes, the government has no repeatable way to assess the technical capability of contractors to determine which has the greatest chance of succeeding. No one knows the state of the practice in the various software domains, e.g., hard real-time, network security. A technology assessment capability would significantly enhance an organization's ability to meet today's increasing demands.

One of the major problems facing software engineering today is the lack of accessible data about the development and use of software products and software development practices and their effectiveness in particular contexts. At present, risk management data are buried within projects and not available to the wider community. Consequently, software engineers are presently forced to resort to non-empirical arguments in deriving or evaluating many software engineering methods and tools.

For example, data from our field work suggest that 67% of identified risks from a broad spectrum of programs arise in only 14 out of a possible 70 risk areas (see Figure 3-21). This provides the empirical basis for concentrating the development of methods and tools that would attack issues in these risk areas before expending the effort to develop methods and tools for the remaining 56 risk areas.

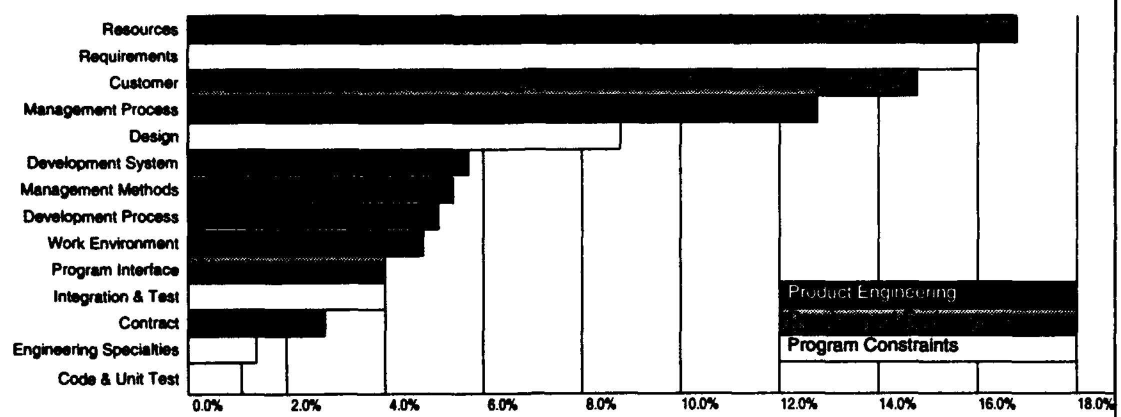


Figure 3-21: Distribution of Risks by Taxonomy Attributes

Such information supports the reason for our major thrusts and points to several community needs: access to risks faced on similar programs and corresponding mitigation strategies, methods to enhance the analysis of risk data, and models and tools to facilitate the process of risk management.

The qualitative methods developed by the SEI have been effective in the field. Quantitative methods in risk management are the next logical step to enhance these methods by providing more precise values for characteristics of risks such as their, likelihood, impact, and time frame. Such increased precision makes the information more visible and "real," particularly to engineers and management.

Finally, tools that are cost-effective and practical would aid the adoption of sound and disciplined risk management practices. Other tools are also required to support the processing of data being collected and organized in the risk repository. The most useful information obtained from the field is verbal and not directly amenable to traditional databases queries. Couple this with the needs of our diverse customer community for using the repository, and we clearly see the need for a database and associated structuring and accessing tools that would work efficiently and well with text-based data.

3.3.4.4 Benefits

Risk management will provide the processes and methods to enable organizations to identify and manage software technical risks within their programs. Using the SEI risk management framework will:

- Improve predictability of results in acquisition and development.
- Improve communications among team members.
- Enable the program to identify and analyze risks.
- Provide a structured process for developing risk mitigation strategies.
- Ensure that program software personnel participate in program risk management.

Incorporating the processes and methods into the acquisition process will provide customers with a basis for evaluating and selecting contractors. Since the process is structured and repeatable, it also provides a foundation for collecting data and looking at trends across multiple acquisitions. The team risk management process will provide customers and developers with the foundation for managing risks in a cooperative, open environment.

Technology assessment will provide organizations with a means to evaluate their technology capability, to formulate strategies for improving technology to mitigate specific risks, to stay abreast of the state of the practice, and to shorten the time for institutionalizing development practices.

Easy access to risk management data will provide program managers (PMs) with the information to make more informed decisions on the technical direction of programs. When all risks become known and managed, then a program can reduce the number of issues that become crises and thus can control and manage routine and critical problems with much improved chances for success.

Simple, widely accepted methods significantly enhance the ability of an organization to analyze its risks and potential mitigation strategies. Qualitative methods serve quite well in many situations, particularly when there are few data on which to base judgment. On the other hand, quantitative methods used appropriately have the advantage of rigor and acceptance in engineering and software development. They also provide greater value in decision making by providing a clearer understanding and discriminating the multiplicity of factors associated with software development risks.

When supported by appropriately automated tools, models can be the basis for codifying theory, principles, and best practices in carrying out all steps required for software development risk management. Hence both the models themselves and the support tools can be significant additions to our arsenal in dealing with software development risks.

In key risk management items, we see the trends shown in Figure 3-22.

3.3.4.5 One-Year Objectives for 1995

Complete the risk action planning process. Risk action planning is a process for developing effective mitigation strategies to reduce program risk. It is based on a problem-solving process and includes methods for selecting the risks that need to be mitigated, detailed analysis techniques, strategy generation, evaluating and selecting strategies, and developing mitigation plans documenting those strategies. Risk action planning builds upon the information developed during risk identification and analysis and provides the risk mitigation plan that includes the metrics for tracking and the basis for control. The objective is to document the process and methods developed and tested in 1994 and 1995.

Key Items	State of practice as of: Today	+2 Years	+4 Years	Impact/Metrics
The SRE method	• no widely accepted mechanisms to deal with program software risks	• systematic application of SRE method	• integrated into overall framework of program management through DSMC	• number of applications • continual use by clients • institutionalized • shows up in contracts
Acquisition Risk Management (ARM) method	• no insight, at source selection level, into program and offeror risks	• concept adopted by DSMC • acquisition community using ARM guidebooks	• accepted as routine practice in all major acquisitions	• ARM used to monitor contract accomplishment
Continuous risk identification methods	• rely on individual knowledge and experience	• systematic methods and tools in use	• integrated into overall framework of program management	• number of vendors • used by clients • institutionalized • shows up in RFPs and proposals
Risk mitigation strategies	• based on past experience and technology • prone to fads and quick fixes	• based on data and systematic methods	• knowledge base linked to database of common risks and strategies (risk prevention)	• number of vendors • used by clients • institutionalized • shows up in RFPs and proposals
Technology assessment	• sporadic or limited use	• Taxonomy-Based Questionnaire method is used to gather state-of-the-practice data	• Questionnaire method used as a tool for technology improvement • EMM used to advance practice	• number of clients using technology assessment • institutionalized by clients
Automated risk management	• bits and pieces with limited coverage and effectiveness	• automated tool use • implement defined methods	• knowledge-based PMs assistant	• number of vendor tools • assessments (evidence of effective tools)
Repository	• limited high-level abstractions of risk and strategies	• prototype database with empirical data • limited access and use	• widespreao access and use • knowledge-based tools to identify risks and strategies • widespread community input of data	• numbers of users • number of successful predictions
Measures	• ad hoc	• risk-driven approach for selection of metrics • quantitative measures adopted	• predictive quantification methods identify risks • accepted as routine practice	• shows up in RFPs and proposals • number of users

Figure 3-22: Trends in Risk Management

Enhance the Taxonomy-Based Questionnaire. The objective is to make the taxonomy-based risk identification process as practical and efficient as possible. The Questionnaire will be improved to cover the system life cycle more completely based on previous field tests and to make it tailorable to recognize program or project characteristics. Based upon this work, oth-

er interview instruments will be developed to determine the state of the practice in various domains, including real-time and network security.

Design and prototype a risk data repository. The design of the repository will be completed and reviewed by selected people from the SEI, government, and industry. Initially, the repository will be populated with data collected from field tests, risk assessments, and other risk technology transition efforts conducted by the SEI and its strategic partners. It will include common risks, risk mitigating actions, results, and lessons learned. In-house tests of an initial prototype version of the repository are planned for 1995. This version will let users describe the status of their software development project by filling out project description templates. The response will be a checklist of concerns or risks encountered in similar projects along with a list of possible mitigating strategies.

3.3.4.6 Work Outputs

Risk Action Planning Guidebook (1995-1996). Risk action planning started in 1994 and will produce a documented process supported by methods in a guidebook format in 1995. A risk action planning training course will be produced in 1996 that will provide 2 days of instruction and interactive hands-on training for the process, methods, tools, and products. By 1996, the process will be expanded to link with the risk data repository to provide an "intelligent aid" to assist clients in developing possible mitigation strategies for a given type of risk and program profile. The process and associated methods are independent of other outputs; however, the linking with the risk data repository for selecting alternative strategies will be dependent on the risk data repository product.

SEI Risk Conference (yearly). The SEI Software Risk Conference addresses the wide range of needs of SEI customers, from practitioners to managers in both the government and civilian sectors. The purpose is to provide a forum for the exchange of ideas, an opportunity to be exposed to best practice, and an awareness of current experiences in software risk management. The conference provides current information on risk management methods, theory, and practice through invited presentations, panel discussions, and workshop formats. It has proven to be an important mechanism in establishing a community of research and practice in software risk management.

Tailorable Taxonomy-Based Questionnaire (1995). To make the risk identification process as practical and efficient as possible, the Questionnaire will be made clearly tailorable. This product will take into account the characteristics of projects being assessed including the domain, life-cycle phase, and type of project.

The first enhancement direction will be to enable the Questionnaire to be easily and efficiently tailored to the specifics of a given software development project. This customizing will enable the project to eliminate questions in the Questionnaire that are not germane to the project due to differences in the life-cycle stage, the application domain, and certain other characteristics such as the use of commercial-off-the-shelf (COTS). The tailorable version of the Questionnaire will be completed in 1995.

The second direction will be to cover, in greater depth, selected areas within the software engineering discipline. Specifically, the Questionnaire will be expanded to cover system performance engineering and system security.

Risk Repository (1995-1996). This effort will assess the feasibility and produce the top-level design of a risk data repository. The feasibility and design will be reviewed to determine whether to proceed. When implemented fully, the risk repository will be a national online resource containing and making available data to develop empirical and model-driven approaches to software engineering. Initially the repository will be populated with data collected from field tests and risk assessments conducted by the SEI and external technical collaborators. A significant and ongoing aspect of this work will be the identification, collection, and organization of project-level data to include not only explicit statements of risks, risk mitigating actions, results, and lessons learned, but also case studies. Data organization and retrieval will be flexible through various kinds of computational analysis including natural language processing (NLP).

The repository will be designed to use more data as they become available. Once obtained, structured, and analyzed, the data will also yield rich information on the relationships among risks, risk causes and attributes, and relative values of risks. The repository will provide information to clients and will become more robust over time as new information is received and validated. For example, users of the repository will be asked to describe the history of their own projects, including their identified concerns and risks as well as the mitigating strategies used and their effectiveness. Figure 3-23 is a schematic of how we envision repository use in risk management. If the prototype warrants, a full implementation will be planned.

The open issues in designing and populating the risk data repository involve structuring these data in a form that will enable appropriate access to the data for all potential users in our customer base. Other supporting issues are:

- The determination of the types of data that are needed.
- The identification of existing and potential sources of these data.
- The methods to acquire existing data.
- The methods to capture data from potential sources (for example, data that are generated in the course of software development but are pragmatically difficult to record).

These issues will be addressed by continuing the current efforts in information analysis and modeling at the SEI and at the Carnegie Mellon University (CMU) Engineering Design Research Center.

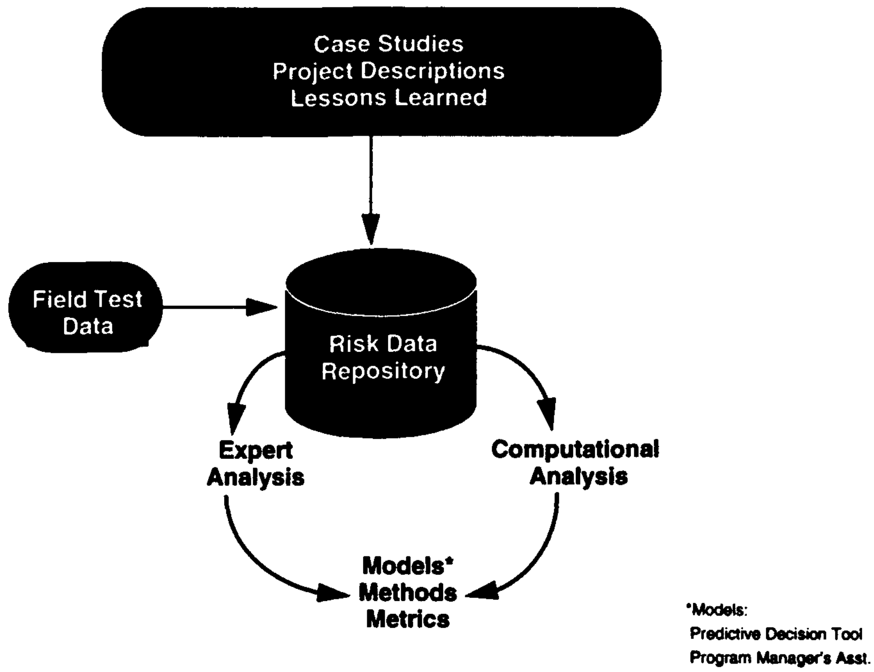


Figure 3-23: Use of the Risk Data Repository

Predictive Decision Tool (1995-1996). Research will begin on a predictive decision model to support SRE. This model will be the foundation for a tool to provide acquisition managers with a way to apply the data resulting from an SRE. The model will use regression analysis to determine possible levels of impact of a risk to a program from multiple causative variables. The variables will come from risk identification methods such as the Taxonomy-Based Questionnaire. Using the regression parameters as weighting factors, the model will be used to predict future outcomes. The ability of the model and supporting tool to accomplish this prediction is dependent on the clear identification of a program's risks and the relative isolation of the development program from major situational changes. Major changes will require new risk identification data to recalibrate the regression model. By reviewing the predictions and coupling them with observations of what actually comes to pass, we expect to identify trend lines to improve the performance of the model, and to publish this information in a technical report.

Program Manager's Assistant (1995-1996). The complexity associated with many software-intensive systems coupled with rapidly evolving technology has resulted in formidable challenges for the software engineering community. These challenges are evident in the need for software engineering practitioners and managers (1) to possess knowledge regarding increasingly broad domains and increasingly complex systems and processes; (2) to share and facil-

itate the sharing of knowledge across diverse specializations; (3) to integrate increasingly voluminous and complex information into the practice of software engineering; and (4) to translate this information into effective design and management decisions. Management of this information (knowledge) is vital for effective software-intensive systems development.

The program manager's assistant is envisioned as a software-based proof-of-concept model to address the issues involved in integrating knowledge into the practice of software engineering management. This integration includes the gathering and structuring of data, its incorporation into the processes, and its conversion into decision-making information for effective engineering and management. The proof-of-concept model would be limited to decisions relating to software-intensive systems development and would focus on a subset of software engineering technologies, e.g. requirements, COTS, models, software maintenance. The approach would build upon SEI experience in field testing, repository development, and risk management technology transition; and in particular, would build upon the team risk management approach to include cooperative work environments.

Through the proof-of-concept model, the project would address the viability of existing knowledge integration and knowledge management models and the effectiveness of their embodiment in computer-based tools for software engineering management. Consequently, this effort would include assessments of the current state of the art in knowledge integration models and tools as applied in software development.

While the focus would be to provide program managers with decision support information for decision making, the perspective will be more general—considering the interrelationship of technology and technology management. It is anticipated that there would be substantial interaction with SEI focus areas that address similar issues relating to software engineering process support, e.g. software information modeling, CASE.

3.3.4.7 Related TO&P Activities

TO&P and technical collaboration customers provide the primary resource to work in real program environments. In some cases, methods are developed collaboratively, but in all cases the field activity provides the testing ground for process, methods, and tools developed for risk management. Major customers include those listed in Section 3.3.4.2.

The field work for SREs provide the testbed for the predictive decision tool, the program manager's assistant, and the empirical data to improve and update the Taxonomy-Based Questionnaire.

The field work such as independent assessments, SREs, and team risk management provides part of the data that would populate the risk repository. Not only are the methods developed by the SEI being proved in field conditions, but also the best practices are being merged and integrated into a total risk management framework for the entire software acquisition and development community. This work is a vital part of the transition strategy for rapidly improving the state of the practice of risk management.

3.3.5 Proposed Add-On Activities

The following section describes proposed core add-on activities in the risk management focus area. The SEI is asking selected members of the software community to evaluate the relative attractiveness of these add-on proposals as possible elements of the final 1995 core program, as funding permits. Appendix A lists all baseline and add-on items.

3.3.5.1 Risk Management in Systems Acquisition and Development

RM-1A Technology Alignment Guidelines

With the downsizing of the DoD, there is a need to combine, simplify, and eliminate support infrastructure. A useful approach is to combine overlapping depot maintenance activities and leverage technology to increase productivity. Technology Alignment Guidelines provide guidance to consolidate and streamline an organization's technical assets. For example, government maintenance facilities have multiple systems, each with multiple versions to maintain, modify, and extend the system life cycle. A risk-driven approach to consolidating and introducing new technology better equips organizations to rationally handle the reality of downsizing and organizational reengineering. Potential customers are DoD laboratories, Service maintenance facilities, or large industry organizations with multiple product line responsibility.

The technology alignment guidelines would provide direction to program management and technical personnel on how to align multiple software-based products and processes into a consistent strategic technical approach. The guidelines would relate the transition of software engineering technology and processes to downsizing and organizational reengineering issues within organizations. The guidance would deal with issues of reengineering, education and awareness, innovation strategies, systems interfaces, implementation, and transition.

RM-2A Software Risk Assessment for Manufacturing

Computer-integrated manufacturing systems are increasing in complexity, and the use of software to manage that complexity is also increasing. The manufacturing domain encounters many of the same software issues that the avionics domain or C3I domain faces. The SRE process would be tailored to the manufacturing domain to identify and assess risks in the development of those systems and to benefit both the customer and developer. The SRE process would enable customers to evaluate proposals, make better choices regarding tradeoffs of technology, and implement a risk management process. The development community would be able to use the Questionnaire and the evaluation process to assess and manage the risks in the development of their systems. Using a common assessment mechanism would enable the community to access and make use of the SEI data repository.

RM-3A Cost Model Risk Management Method

Managers focus on cost and schedule as barometers for the health of their projects. Cost and schedule issues are often merely symptoms of problems, but they are what PMs understand. There is a strong connection between cost and schedule drivers in estimation models and the risk taxonomy. This effort would synthesize the SEI risk activity with the work at the University of Southern California (COCOMO 2.0). The output of this effort would be a method to identify

the sources of risks and to estimate their cost and schedule. The product would be documented and pilot tested in several scenarios to validate the hypotheses. Pilot testing would be dependent on TO&P. The benefit to PMs would be an improved process for evaluating cost and schedule estimates and basis for making better decisions regarding program plan and risk mitigation strategies.

RM-4A Technology Assessment Taxonomy-Based Questionnaire

The SEI and the software engineering community lack a systematic and repeatable process for assessing risks in technology. We propose to extend the Questionnaire to enable the community to identify technical issues and populate models like the EMM. The process and method would provide a consistent approach for the collection of data upon which to base software engineering improvement strategies (e.g., reengineering, architectures).

There is great need in the community for a method to determine technological capability and to formulate a coherent improvement plan to enhance an organization's ability to deliver quality products within budget and schedule. Data would be gathered on the state-of-the-practice of software engineering technology to assess an organization's technology capability. Comparing the global state-of-the-practice with the state-of-the-art would reveal appropriate technical approaches in need of transition into the community.

This effort would first develop interview instruments to determine the state of the practice in various domains. The data gathered from those interviews would become the foundation of this effort, which would include extending the Questionnaire and testing the hypothesis that the Questionnaire method can be extended to cover technology assessments. To enable a concrete and operationally useful test of the hypothesis, the immediate area of application of the Questionnaire would be aiding the development of the SEI EMM. This effort would include constructing an interviewing instrument and conducting interviews to determine the state-of-the-practice, reformulating the Questionnaire based upon the interview data to assess the state of the practice, and conducting pilot tests in the community. The data gathered would also feed directly into the risk repository and provide a database of technology information for the entire software community.

RM-5A SRE Train-the-Trainer Course

The SRE is the basis for assessing and analyzing risks for a project and is the foundation for implementing a risk management process. The SRE process has been successfully pilot tested and used by numerous clients. As the SEI has transitioned the SRE to client organizations, these organizations have requested a training course to allow them to teach others in the organization how to conduct SREs. This train-the-trainer course is required to sustain the SRE process inside an organization after the SEI has completed the initial transition effort.

The output from this effort would be a set of documents, training videos, course materials, and a training guide for conducting the SRE training.

RM-6A State-of-the-Practice Report

The state of the practice for risk management is not well documented. The SEI has data regarding the state of the practice based on conducting risk assessments for sponsors and clients, but not necessarily indicative of the wider software acquisition and development community. This effort would enable the collection of information on the state of the practice across multiple domains in both the government and commercial sectors. This could lead to a periodic update (e.g., biannual). Information would include state-of-the-practice statistics, observations, and best practice. This information would be useful for measuring and comparing clients as well as for indicating the possible impact of SEI transition activities. This information would also be beneficial for developing and implementing risk management improvement efforts with SEI clients and sponsors.

RM-7A Risk Management Key Practice

The CMM key practice area of risk management is needed to further strengthen the CMM such that organizations can adopt these practices into their process improvement programs. This key practice area was purposely omitted from early versions of the CMM because standard practice was undefined. This task would fill the gap in the CMM by identifying and defining risk management as a key practice. This work will be synchronized with other maturity model definition efforts within the SEI (e.g., the SECMM). This work is dependent on the corresponding maturity models affected (at a minimum, CMM and SECMM). This work began at a very low level in 1994 by establishing a working review group to initiate collaboration.

RM-8A Acquisition Risk Management Guidelines

Acquisition PMs do not have a well documented set of methods or processes for identifying or assessing software risks as part of the source selection process. The acquisition community has applied the SRE to program management and used the resultant mitigation strategies to their best advantage in the allocation of resources across their programs. The expectation of the acquisition community is that these risk methods would naturally migrate, where applicable, into the actual "buying" phase of the acquisition life cycle. This work uses risk identification, analysis, and mitigation strategies during the "buying" phase of the acquisition life cycle to ensure acquiring the best system possible. The initial steps included the development of three Army guide books for the acquisition authority, the acquisition source selection evaluation board, and the proposal/bidder. This work would extend the product across the services, develop a training process for the reviewers, and pilot test the process across the services. This effort would be augmented by TO&P for the pilot testing.

RM-9A Software Acquisition Capability Maturity Model (SACMM)

The government has a need to assess the maturity of its internal software acquisition management process. To address this need the SACMM would be developed based on the SEI CMM and risk management taxonomy.

The SACMM would provide a principled, public model for appraising software acquisition maturity. It would be generic so that it can be used by any organization acquiring software. This includes PEOs who may be acquiring software across several projects, PMs who are responsible for single system acquisition, or software support activities responsible for supporting PEOs and PMs. This model would also include provisions for the participation of other functional organizations involved, such as test, product assurance, and laboratories. Industrial acquisition organizations may also use the SACMM for improvement. The SACMM would not address the system-level acquisition process, but rather be an adjunct to it.

The SACMM would promote process improvement in software acquisition. This is especially important since software developers are well into improving their capability maturity through use of the CMM. As software developers' processes improve over time, they will begin to overwhelm those organizations acquiring software since they will not be able to interface effectively with software development organizations. Software acquisition maturity must keep pace with software development maturity so that software acquisition organizations are in a position to capitalize on the improvements achieved in the process of software development. Software acquisition organizations need to improve their capability maturity so that they can acquire higher quality products cost effectively. The SACMM would provide software acquisition organizations with many of the concepts needed to improve their processes and keep pace with software development organizations.

The SACMM, like the CMM, would define KPAs for each of its five levels of maturity. The KPAs define the requirements that must be satisfied to accomplish that level of maturity. In other words, progress is made in stages or steps. The levels of maturity and their KPAs thus would provide a road map for attaining ever higher levels of maturity. The SEI's extensive experience in developing risk, SECMM, and the CMM is directly applicable to developing the SACMM, since the three models must be synergistic.

3.4 Disciplined Engineering

Demand for software-intensive systems is constantly increasing, and these systems are growing in both size and complexity. More effective and efficient software engineering practices must be employed to cope with the problems created by increased demand, size, and complexity.

Successful practices in mature engineering disciplines include the use of system architectures and models, and systematic analysis of system properties and their tradeoffs. In mature engineering disciplines, engineers systematically make choices, synthesize, and evolve systems in a predictable manner. The desired state of software engineering is therefore to become an engineering discipline by:

- Using architectures and models to represent views of the system for synthesis and analysis.
- Analyzing tradeoffs of system properties to improve predictability and control of quality attributes.
- Automating the practices to increase productivity and reduce human errors.

We refer to this state of software engineering as model-based software engineering (MBSE) (see Figure 3-24).

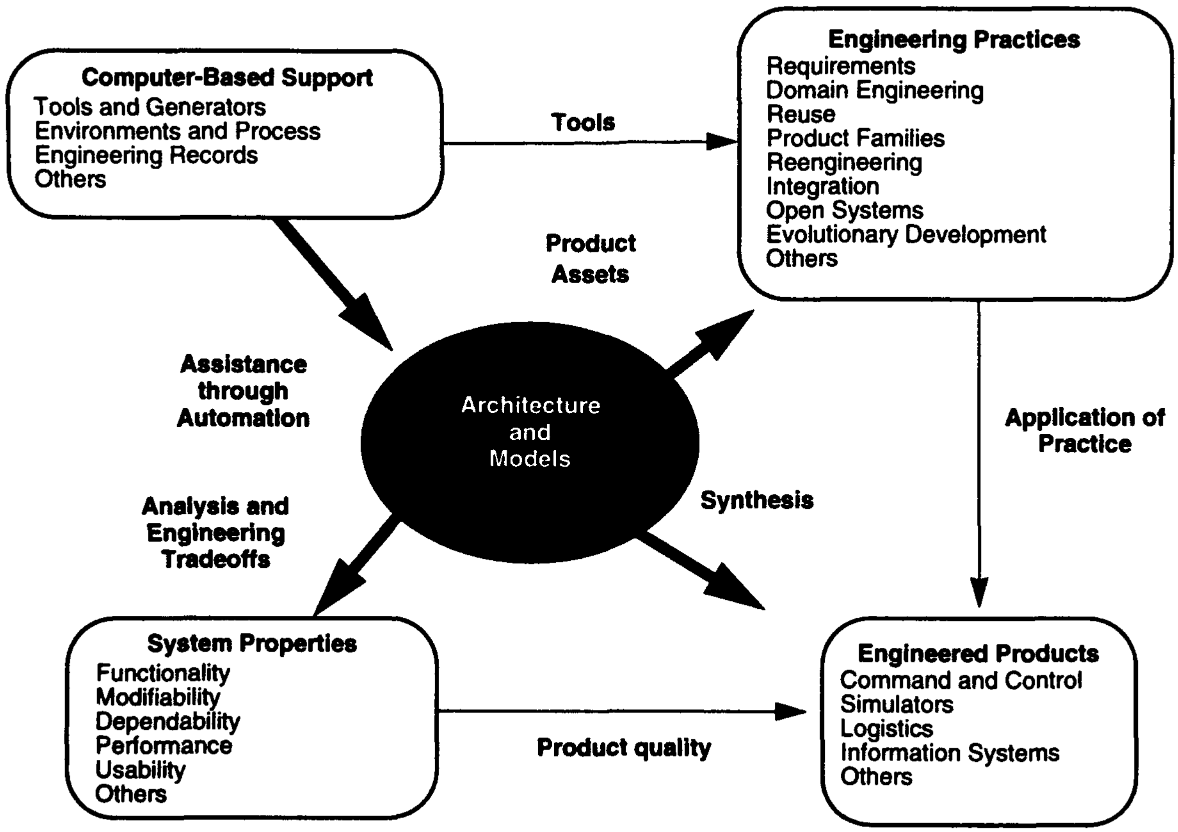


Figure 3-24: Model-Based Software Engineering (MBSE) Practice

In MBSE, architectures and models play a central role in the life cycle of software-intensive systems. Libraries of architectures and models can be assets for various engineering practices, can drive the synthesis of products, and can be analyzed with respect to various desirable or required system properties. Achieving the desired state requires baselining and improving the current state of engineering technical practice by focusing on surveys and methods for assessing existing and emerging technologies, architectures, and models (see Figure 3-25).

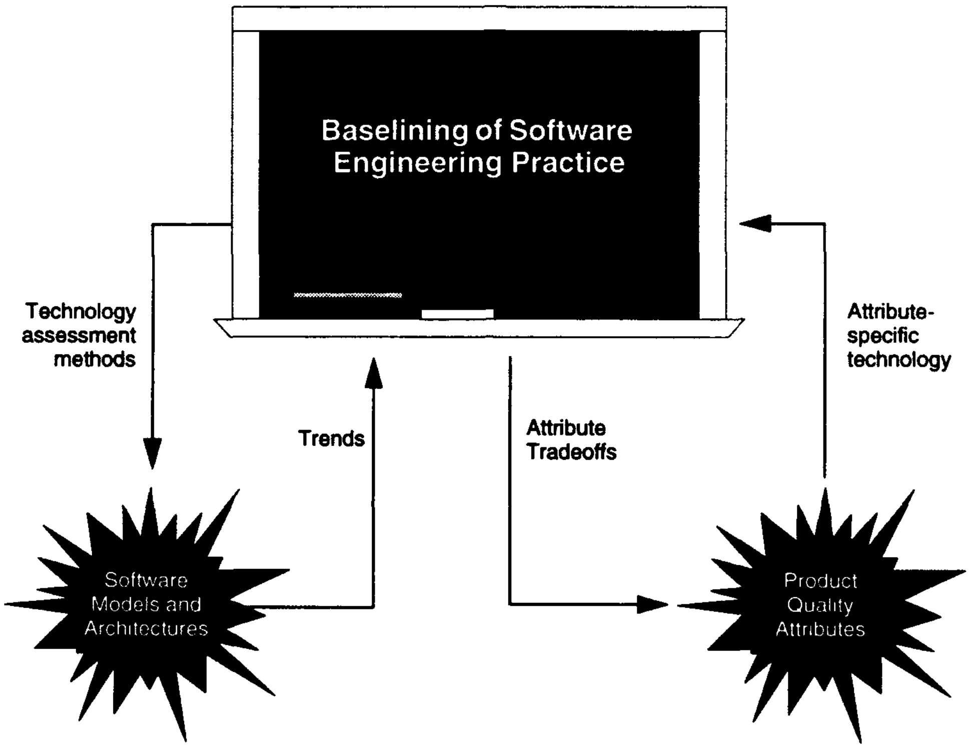


Figure 3-25: Baselining the State of Engineering Practice

Improving the practice has two major components in which we are active: **technology maturation** and **technology adoption**. Technology maturation focuses on maturation and evaluation of software engineering practice through technology. This contributes to improvement of best practice. This component, illustrated in Figure 3-26, focuses on technology evaluation, identification of technology trends and roadmaps, and the adaptation of MBSE practice to incorporate these advances.

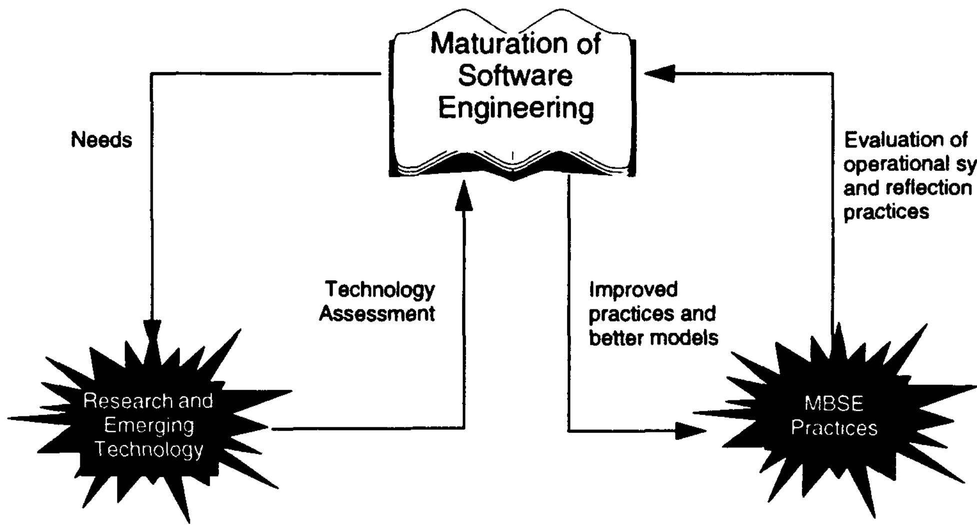


Figure 3-26: Improvement of Best Practice

In general terms, this maturation of practice is patterned after other engineering disciplines that are built on a body of engineering knowledge based on theoretical principles or empirical results. That body of knowledge exists as a set of engineering practices. The mission of the SEI is to improve the practices (managerial and technical) of software engineering. Improving the technical state of the practice requires:

- A characterization of what improvement means.
- A mechanism for describing the state of technical practice.
- A roadmap for improvements in the state of the practice.
- Application of these concepts to specific areas.

A feasibility study conducted in 1993 developed an EMM to describe a notion of maturity that can be used to characterize what improvement means for the technical aspects of software development. The EMM characterizes improvement in terms of the ability to predict and control software quality attributes.

The EMM characterizes the state of technology along three fundamental components:

- Engineering knowledge
- Engineering methods
- Engineering tools

Maturing the knowledge base is the fundamental enabler for maturing the state of technical practice. Figure 3-27 shows that tools, practices, and knowledge have an impact on one another, with the levels of knowledge maturity (described below) in the center:

- **Folklore:** Pragmatic rules for recurring technical problems exist as an informal body of undocumented knowledge.
- **Descriptive:** Problem characteristics and associated solution techniques are categorized and documented.
- **Measured:** Useful metrics for characterizing artifact properties exist, and an empirical understanding of the relationships between parameters emerges.
- **Understood:** Theoretical models exist that are useful in predicting and explaining artifact behavior.
- **Refining**: Theoretical models are refined and their applicability broadened.

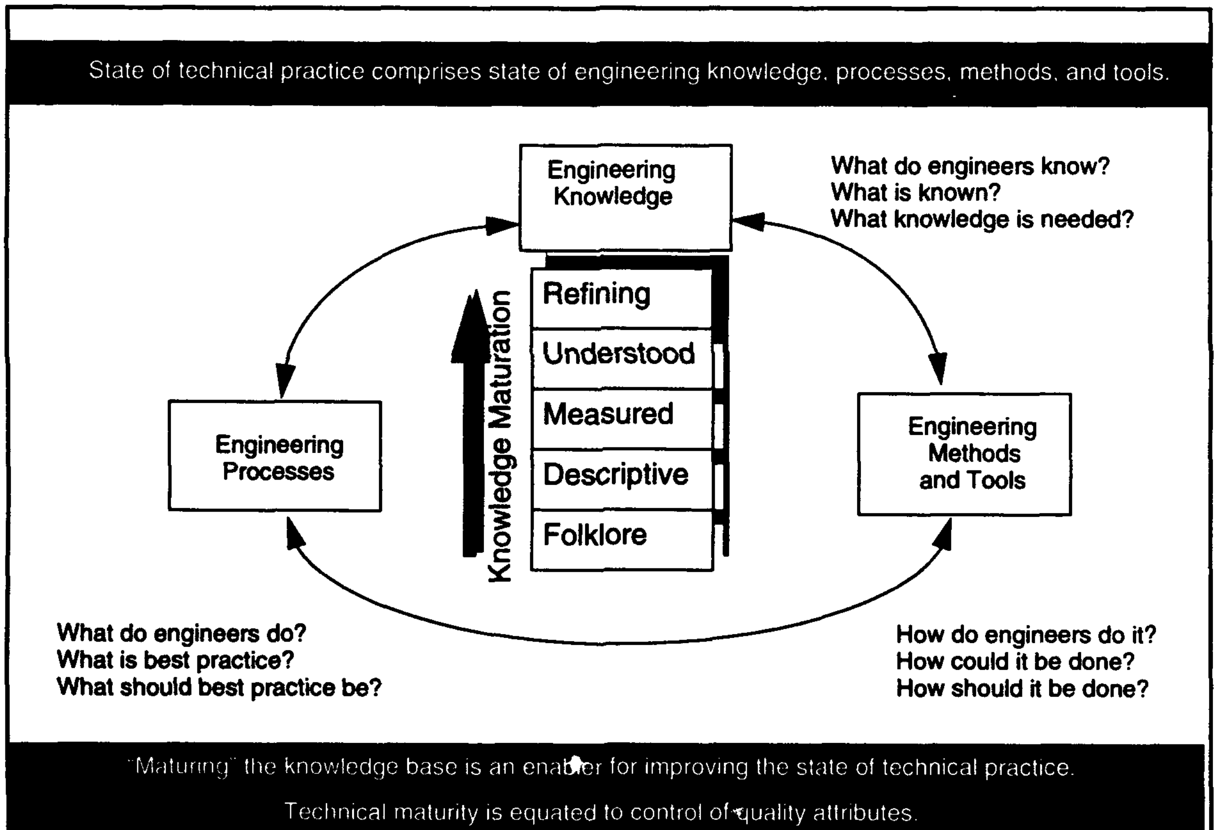


Figure 3-27: An Engineering Maturity Model (EMM)

A primary goal of developing an EMM is to create a vision of technology maturation that is shared among practitioners, researchers, and tool vendors. Practitioners could identify areas in which knowledge needs to be matured, researchers could work on problems that are both

challenging and important for practitioners, and tool vendors could build tools in support of mature knowledge and practice.

Technology adoption focuses on improvement of current practice, i.e., getting improved best practice adopted by the majority of the practitioner community. As a result, this component emphasizes improving organizational capability—in the case of this focus area, technical engineering capability as compared to management capability. Our focus regarding this component can be summarized as leveraged transition, illustrated in Figure 3-28.

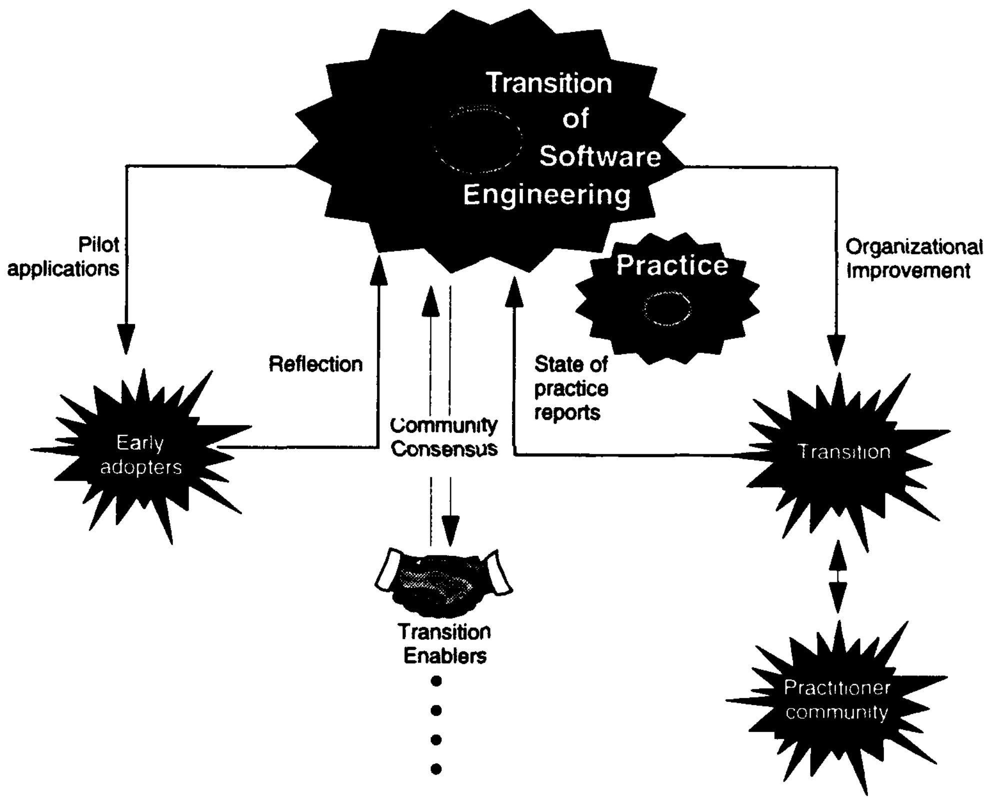
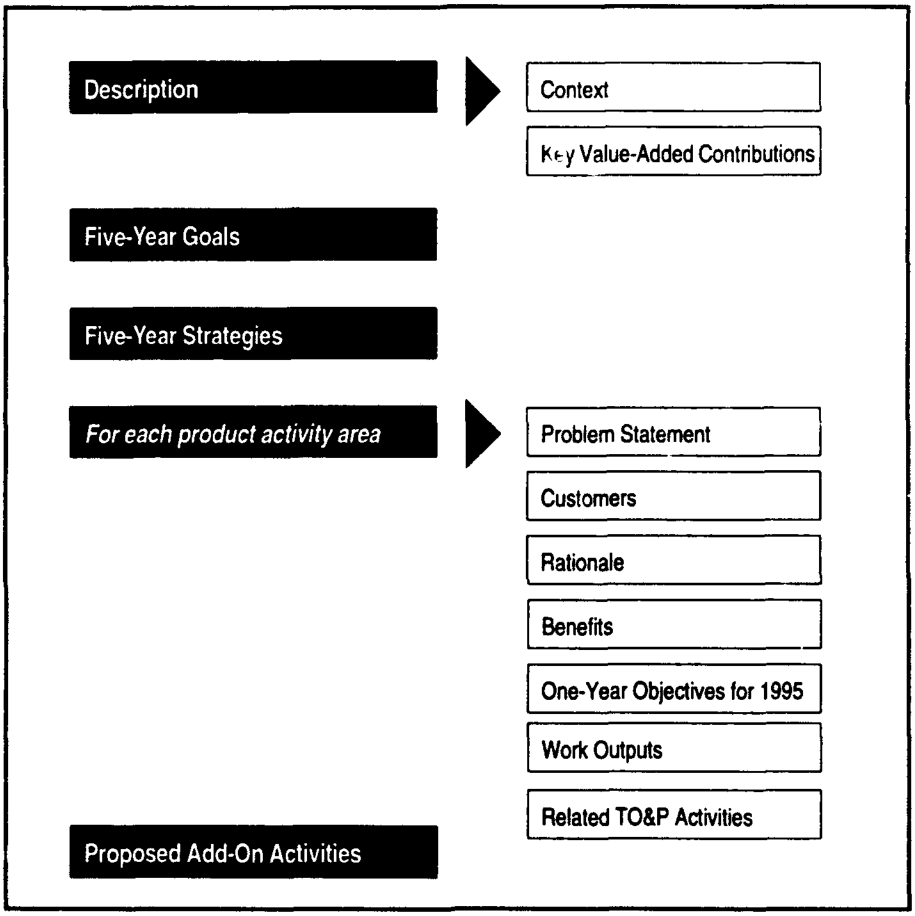


Figure 3-28: Improving the Practice

The approach to improving practice through leveraged transition involves case studies of existing best practices; pilot application of promising technology with our TO&P customers in application domains such as flight simulators, command and control systems, and SEEs; and dissemination of lessons learned from the resulting improved practice, e.g., in the form of a tailorable course collection for MBSE. It also involves working with transition enablers such as standardization forums, technology interest groups in government and industry, policy groups, and SEI strategic partners to build community consensus. Finally, it involves cooperation with

owners of the transition infrastructure to evolve strategies for leveraged use of this existing infrastructure to transition improved best practice to improve the practitioner majority. These owners include policy groups and their implementors, strategic decision makers in organizations for adoption and improvement of practices, and leaders in the educational community.

The sections for this focus area are:



3.4.1 Description

3.4.1.1 Context

The focus on disciplined engineering enables the SEI to maintain its strategic activity in maturing the technology. The results are intended to reduce technical risk in software development and, depending on the target audience, might be introduced directly (e.g., through early adopters) or as part of a risk mitigation strategy.

This focus area is the result of combining work from two existing, complementary focus areas. The first area, methods and tools or software engineering techniques (SET), focused on identification, development, evaluation, and transition of technologies for architectures and domain models for software-intensive systems. The second area, real-time distributed systems or product attribute engineering (PAE), focused on identification, development, evaluation, and transition of technologies to predict and control the quality attributes of software-intensive systems.

The combined focus area addresses four product activity areas, described below. Figure 3-29 illustrates how the activities in the former focus areas contribute to the four new product activity areas.

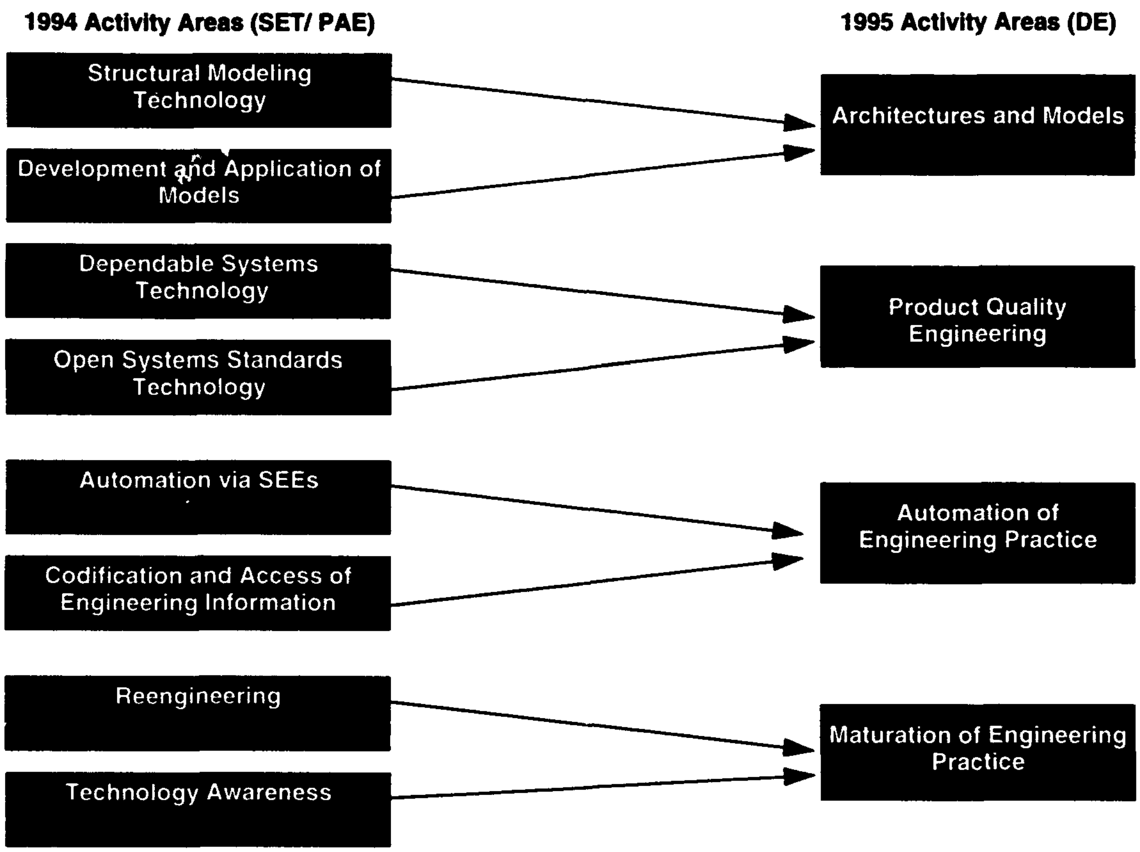


Figure 3-29: Mapping of Activity Areas

3.4.1.2 Key Value-Added Contributions

The SEI, in cooperation with government agencies (NIST, Office of the Secretary of Defense [OSD], ARPA, DISA, and the services) and other organizations (ISO and IEEE Computer Society technical committees and industry groups) establishes a shared framework for evaluation and measurement of technology. Assessment of technologies results in technology trend

analysis and technology roadmaps. Evolution of evaluation methods will result in more quantitative analysis and prediction of system quality.

Specific contributions of this focus area are described below in terms of the four product activity areas. In addition to specific outputs listed under each of the activity areas, the focus area will develop white papers, briefings, and prototypes addressing technical concerns of our sponsors and the software engineering community in general.

Software Architectures and Models. This product activity area focuses on analyzable representations of software-intensive systems and on technology for creating, using, and evolving such representations. The SEI facilitates community understanding of the concepts behind architectures and models and the complementary nature of various government and industry efforts in this area. In cooperation with external parties, the SEI identifies current practice in evaluating software architectures, in particular from the acquisition perspective. This baseline is the starting point for improving the practice of software architecture evaluation. The improved practice uses quantitative methods to analyze the tradeoff of quality attributes in proposed software systems solutions.

The SEI evolves a framework for architecture technology evaluation. This framework encompasses criteria and methods for assessing the effectiveness of existing (commercial) and emerging (research) architecture representations, languages, and tools for analysis and synthesis. The application of this framework allows us to identify technology trends and maintain technology roadmaps. The improved practice allows practitioners to make choices as to the most appropriate technology for their purpose.

The SEI evolves a framework for system representation based on architectures and domain models. This framework encompasses existing best practice in reuse-based, domain-specific software engineering as well as emerging technologies with the potential for improving this practice. This framework can be used by practitioners to make appropriate choices in methods and tools to adapt the engineering process to the specific project or organization. The improved practice encourages effective systematic reuse in the form of product families and product-line engineering.

The SEI fosters community-accepted reference models for architectures in various application domains. As a neutral party with no special interests, the SEI is in an ideal position to investigate and promote technological solutions that use a combination of available technology such as dependable systems and open systems. Currently, the SEI is actively involved in domains such as flight simulators, logistics systems, and integrated environments.

Product Quality Engineering. This product activity area focuses on prediction and control of quality attributes of systems and the tradeoff of these attributes across alternative designs. Researchers focus on individual attributes (e.g., reliability, security, fault-tolerance) and seldom look at combinations of attributes, while practitioners need guidance for selecting congruent technologies. For selected groups of quality attributes, the SEI, in collaboration with external

partners, will identify and demonstrate the range of possible tradeoffs and make the results available to practitioners in guides to best practice and roadmaps. The improved practice provides predictability and control of product quality.

Mature architectures provide predictability and control over desired combinations of attributes. Developers and researchers focus on specific architectures and architecture description languages and seldom look at alternatives. The SEI will survey and analyze existing and proposed architectures and provide models for selecting architectures to meet desired combinations of quality requirements.

Automation of Engineering Practice. This product activity area focuses on computer-based tool support to perform informed engineering decisions. This activity area addresses development and maintenance activities and access to the information needed to support them. Technology developers focus on promoting their own solutions, but users do not have the luxury of drawing on lessons from other domains to assess the benefits and limitations of proposed and emerging technology. The SEI evolves a technology adoption framework for CASE environments. This work takes place in the context of an Institute of Electrical and Electronic Engineers Inc. (IEEE) and ISO standards activity, and we are validating it with case studies. Improved practices allow organizations to reduce the risk involved in adopting new CASE environments.

In cooperation with government (NIST, ARPA) and industry partners, the SEI evolves an evaluation framework for environment technology and maintains technology roadmaps as well as methods for choosing technology most appropriate to the particular need. The improved practice accelerates process automation in SEEs by building on lessons from office automation, tutoring systems, expert systems, and existing process-centered environments and tools. The adoption framework builds on the evaluation framework for environment technology and brings to bear systematic ways of dealing with non-technical issues. The SEI is in a position to leverage expertise and experience, both in-house and externally, in addressing these technical and non-technical issues.

To make informed decisions, practitioners need effective access to software engineering information and technology to support it. Software engineering information falls into two categories: information about the system being produced or maintained—sometimes referred to as the designer record of a system—and information about the practice of engineering software systems, in the form of an interactive software engineering handbook. The increasing amount of information results in information overload unless appropriate technology support makes the information easily accessible on demand. A range of technologies is becoming available, including hypermedia, multimedia, natural language, intelligent learning, and tutoring. The SEI is in a unique position to evaluate this technology, and apply it in the context of software engineering by leveraging the CMU focus on this technology as a key research area.

Maturation of Engineering Practice. This product activity area focuses on baselining and improving engineering of software-intensive systems. This activity area has a strong customer view and addresses technical and non-technical issues drawing on insights from other focus areas and the community. We currently focus on performance engineering, reengineering, domain engineering, integrated environments, and engineering of flight simulators.

In performance engineering, the SEI performs case studies of performance tradeoffs and improvements by today's practitioners. In addition, the SEI assesses technologies in support of performance engineering using the EMM as a framework to characterize increasing maturity of performance engineering knowledge and practice.

In reengineering, the SEI acts as a central point for lessons learned in reengineering practice. In cooperation with the community, we evolve a framework to assess and improve systematic evolution of legacy systems. A guide to best practice reflects common understanding of the community of best practice and the full range of issues to be concerned with for successful evolution of legacy systems. In addition, the SEI transitions advances in domain modeling, representation of architectures, and tools for analysis of quality attributes to the reengineering community. The SEI also accelerates advances in design records and decision-tree concepts to improve the practice.

In domain engineering, the SEI provides a framework for MBSE, allowing practitioners to choose appropriate technology and models. The SEI complements this technical framework with a framework for business case and adoption strategy development.

In integrated environments, the SEI is leading efforts for community consensus on a reference model and a federated architecture supporting heterogeneous solutions as a basis for engineered environments. This effort is complemented with a cooperative initiative to provide quantitative methods to engineering and community understanding of an adoption framework for leveraging standards organizations.

In flight simulators, the SEI works with strategic partners to gain acceptance by the flight simulator community for a common reference model and architecture. In this context, the SEI has been involved in addressing the implications of the acquisition process.

3.4.2 Five-Year Goals

The overall goal of this focus area is to mature the software technologies and engineering processes used in the disciplined engineering of software-intensive systems. The effectiveness and efficiency of engineering software-intensive systems are improved by building a body of software engineering knowledge based on experience and scientific insight, and by reflecting

it in product quality attributes, product architectures and models, engineering practices, and automation of practice. This translates into four subgoals characterized by the attainment of:

1. Commonly agreed-on measures—identified parameters of importance (factors) and techniques for measuring those parameters—of product quality attributes and technology for product quality improvement.
2. Methods for quantitative analysis of software architectures resulting in improved predictability and control of product quality attributes.
3. Commonly agreed-on software engineering practices based on domain models and system family architectures (as evidenced in product-line engineering) resulting in measurable improvements of software engineering technical practice.
4. Automation of software engineering practices through computer-based support, resulting in more efficient software engineering activities, managing of information, and routine installation and effective use.

The disciplined engineering focus area will use an advisory board to provide advocacy and community feedback on strategy and outputs. The board will be comprised of respected members of the software engineering community who are knowledgeable about disciplined engineering. The members will be drawn from industry, government, and academia, as follows:

- Industry: Drawn from different corporations within the defense and aerospace sector or from different corporations within the commercial sector.
- Government: Drawn from different branches of the armed services and different agencies.
- Academia: Drawn from different universities. Having one member from CMU will be encouraged.

3.4.3 Five-Year Strategies

These goals will be pursued in terms of an MBSE practice, demonstrated in the context of selected perspectives such as reengineering and domain engineering.

- **Baselining software engineering practice**: Identify a framework for characterizing current software engineering practice and examine the current state of practice through selected case studies and lessons learned in cooperation with the community. The results will be reflected in guides to best practice and will be the baseline into which promising technology will be inserted to improve the practice.

- **Maturing software engineering practice**: Identify key technology areas that have the most potential for improving the practice (value-added), and foster their maturation and adoption by the software engineering profession. The results will be reflected in technology trend analyses and roadmaps, pilot application, and lessons learned. The methods for assessment of software engineering technologies with respect to their value-added will be the basis for practitioners making choices when putting their practice into operation. The EMM provides the general framework for guiding this strategy.

- **Leveraging transition of software engineering practice:** Obtain community buy-in and build community consensus through strategic partners with high-leverage potential. The result is a transition and institutionalization infrastructure based on early adopters, transition enablers such as standardization groups, and other relevant organizations.

The three strategies apply across all four goals. In each of the four product activity areas, elements of the strategy have been applied and can be built upon.

3.4.4 Software Architectures and Models

This section discusses the product activity area related to software architectures and models.

3.4.4.1 Problem Statement

The engineering of software-intensive systems requires a disciplined approach for handling complexity. Software architectures and models have been recognized as a way of addressing understanding, analysis, and evolution of systems. A number of technologies exist to support this approach to varying degrees. However, there is little community consensus on:

- What is an appropriate representation of a software architecture.
- How to select technologies and combine them to meet the needs of practitioners.
- How to apply technologies effectively.

This product activity area is concerned with software architectures and models in various domains as a basis for understanding systems, recognizing commonality and variability across similar systems, and engineering of system families. One aspect is the evaluation of existing and emerging technologies for their maturity and effectiveness in support of the software architectures.

A second aspect is the evolution of current practice to incorporate this technology. MBSE serves as a framework for this evolution. Under MBSE, engineering activities can be characterized as domain engineering and application engineering. Domain engineering refers to the creation of domain models and domain-specific reference architectures. These outputs are the basis for application engineering of individual systems.

Domain and application engineering are, in turn, supported by architecture, domain analysis, product quality attributes, and other related technologies. These technologies provide the abstraction, analysis, and synthesis techniques that form the basis for domain and application engineering.

Figure 3-30 illustrates the domain engineering and application engineering approaches, moving from left to right across the graphic. The technology areas to support requirements, architecture, and implementation techniques are shown orthogonal to the two engineering approaches. These technology areas include techniques for architectural synthesis and analysis, domain analysis, object-oriented technology, performance analysis, and quality assess-

ment. Where available, the activity area uses and transitions existing technology. In other cases, existing technology is matured through work within the activity area to advance the state of the practice. Our efforts in this activity area will produce a center of expertise in architectures and models to support state-of-the-art and state-of-the-practice technology.

3.4.4.2 Customers

Customers with technology interests include:

- DoD-sponsored programs: e.g., Software Technology for Adaptable, Reliable Systems (STARS), DSSA, Prototech, and Central Archive for Reusable Defense Software (CARDS).
- Development organizations wanting to build corporate competency and assets in architectures and models.
- The DoD acquisition community evolving to architecture- and reuse-based approaches.

Customers with domain interests include:

- Training simulator community
- Command and control community
- Tool builders

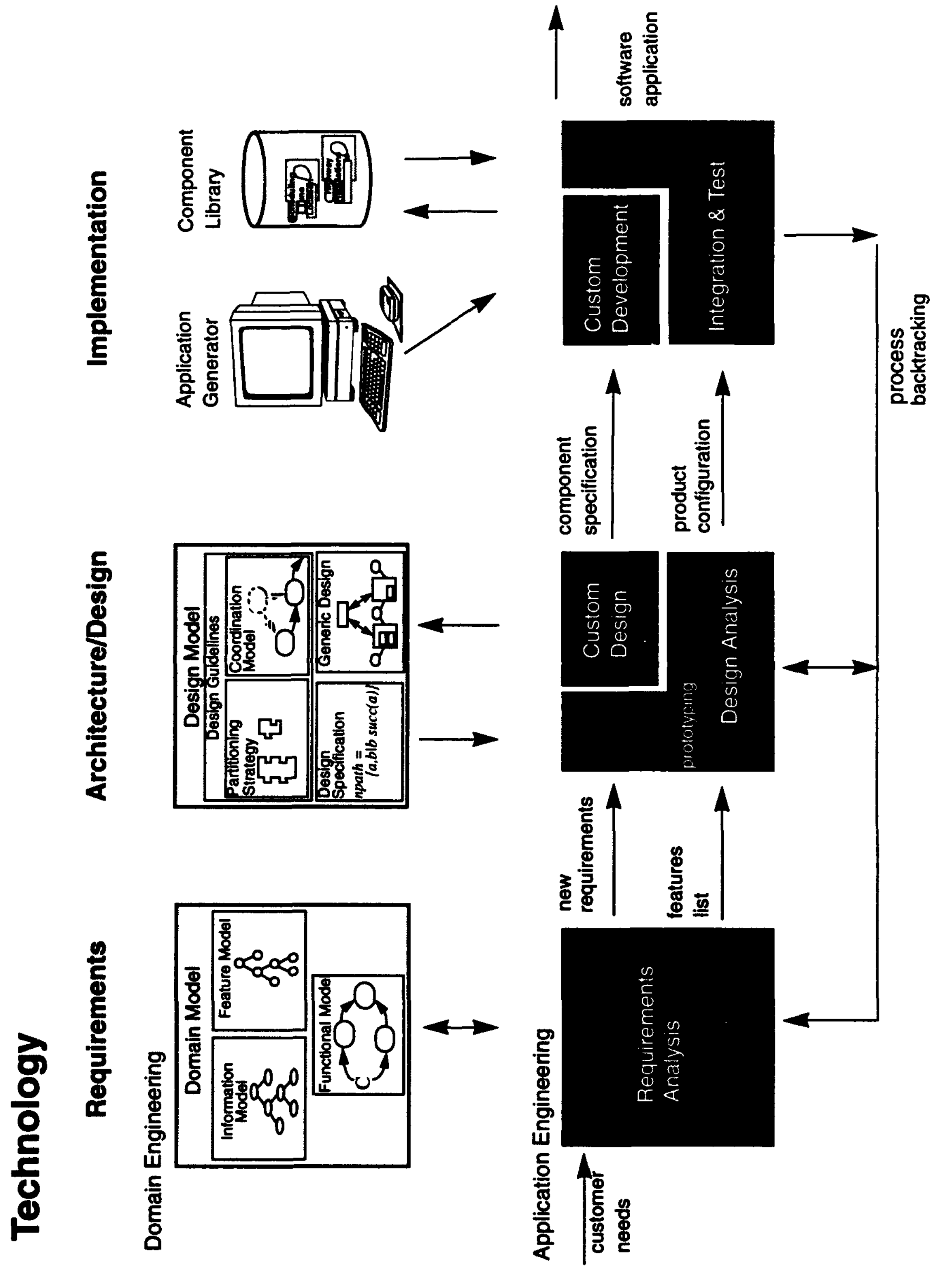


Figure 3-30: MBSE Leverage to Support Domain and Application Engineering

3.4.4.3 Rationale

The need for this activity area is seen in the demand for development approaches emphasizing architecture- and reuse-based technology.

- Other engineering disciplines recognize architectures as a key component for managing complexity. Similarly, other engineering disciplines use models as the appropriate level of abstraction for reuse. Yet techniques for effective evaluation and analysis of architectures and models, and synthesis of systems based on architectures and models, are not widely understood or known.
- Successful program and product planning can occur only when related architecture and reuse activities are brought together to share technology and advance the engineering practice. New architecture technologies continue to emerge, and we must be able to build on related efforts. This must be a consensus-building process between different parties, be they system developer or customer, product-line manager, or parties in a software component industry.
- The effects that particular design decisions have on system qualities often cannot be determined in a quantitative or repeatable way. Developing systematic ways to relate the quality attributes expressed by a system to the system's architecture provides a sound basis for making objective decisions about design tradeoffs and enables engineers to make reasonably accurate predictions about a system's attributes that are free from bias and hidden assumptions.
- Patterns discovered during a domain analysis can be effectively used to synthesize a reference architecture for a domain. These patterns of function, form, and coordination capture fundamental abstractions about the domain that, when realized in a pattern-based architecture, significantly reduce the complexity of the system. These architectures provide high leverage in the areas of understandability, integration cost and schedule, and maintenance cost. Current synthesis techniques do not adequately address how to use these patterns in an architecture.
- Specific studies point to the need for increased effort in this activity area, for example, the Government Accounting Office (GAO) reuse report (January 1993) and the National Research Council report, "Software Engineering: Scaling Up," (1991).

3.4.4.4 Benefits

This product activity area will provide significant benefit to customers by:

- Fostering shared views among software professionals of appropriate architectures and models for similar systems and system functions.
- Evaluating the maturity and effectiveness of architectural representations and their analysis capabilities on actual systems.
- Supporting the ability of systems engineers to make informed tradeoffs regarding quality attributes of systems.
- Linking domain and application engineering to provide a systematic approach to gain the benefits promised by reuse.
- Demonstrating abstraction techniques (e.g., domain analysis, architectural modeling) in support of engineering of software-intensive systems.

- Predicting and controlling product attributes through architecture-level analysis and synthesis.
- Addressing the needs of integrated product development teams for product families through a framework for analyzing existing development practice.

Figure 3-31 illustrates the trends seen in this area over the next four years.

Key Items	State of practice as of:			Impact/Metrics
	Today	+2 Years	+4 Years	
Domain engineering	piloting tutorial/work-shop	best practice report in place; standard training in place	guidebook and training capability transitioned	number of vendors used by clients institutionalized
Community standardization for architectures and models	examples; no common terminology	consensus among experts; technology roadmap	adoption handbook business case	used by clients; institutionalized; shows up in RFPs and proposals
Evaluation of architecture technology	machine-processable architecture representation	quantitative analysis of architectural description	composition and evolution based on architectural descriptions	demonstrate productivity gain 10X and more
Business case development for MBSE	defining report community interest	examples; piloting training	best practice report (e.g., decision aids); standard training in place	organizations implementing MBSE seen as corporate reengineering tool
Architectures and models technology base	flight simulators C2 tactical decision aids others specific to the organization	information and other real-time software systems added guide and demonstrations for development environments	multiple domains mature tool support incorporation into development environments	architectures and models as basis for development; vendor support

Figure 3-31: Trends in Software Architectures and Models (Continued)

3.4.4.5 One-Year Objectives for 1995

- Evaluate software architecture- and model-based approaches in terms of suitability for transition; and promote architecture and modeling processes, methods, and tools as a key technology for software engineering.
- Define and provide examples of accepted domain models and architectures to form a "corporate memory" for effectiveness of state-of-the-art and state-of-the-practice architecture technology. This will give practitioners an understanding of the relationship between architecture, risk, and engineering knowledge.
- Create a common universe of discourse regarding software architectures to promote understanding, interchange, and identification of areas requiring additional research. Through this role, we hope to induce a paradigm shift for the community.

- Provide practitioners with a framework for understanding technology and for supporting the capability to make disciplined, informed choices regarding the selection and adoption of technologies.
- Provide practitioners with a framework for MBSE practices.
- Develop business case for systematic reuse. This includes:
  - Baselining existing practice.
  - Creating methods to help organizations assess their ability to transition to systematic reuse methods based on technology and ROI estimates.
  - Helping organizations define the products they need to support these reuse methods and estimate costs to develop them.
- Produce quantitative techniques for evaluating impacts of design alternatives on system qualities. Developers will recognize how to use these techniques from early in the development process.

Several proposed add-on activities provide additional leverage to the baseline investment. DE-1A, DE-2A, and DE-5A focus on evaluation of technology; DE-4A and DE-6A focus on advancement of technology; and DE-11A and DE-13A focus on understanding of technology and its impact (see Sections 3.4.8.1 through 3.4.8.4).

3.4.4.6 Work Outputs

Guide to Best Model-Based Software Engineering (MBSE) Practice: Edition 1 (1995). This guide discussed best engineering practice based on domain models and domain-specific architectures, spanning the full life cycle, and addressing reuse as well as strategies for adoption of such technologies. The guide represents one of the foundations of the program and builds on core and TO&P investment in MBSE in 1994 and earlier.

Guide to Software Architecture Assessment Practice (1995, 1996). This guide discusses how expert practitioners evaluate a software architecture to determine its "goodness." The guide will address issues of identification, representation, and investigation of architectural information, system properties, rules of thumb for assessing their presence, implications on the quality of the resulting system, and methods for systematically making judgment calls.

This activity involves case studies of expert teams in the practitioner community. It has been seeded for 1994. In 1995 we intend to offer a workshop to evolve and validate community understanding of the architecture evaluation framework reflecting current practice. In 1996 we will publish a report on current evaluation practices and outline improvements of this practice through maturing software architecture representation languages identified in a companion activity (see Section 3.4.8.1, DE-1A).

3.4.4.7 Related TO&P Activities

ASC/YT is sponsoring the development of structural modeling technology for aircraft simulators (guidebook and core architecture for flight simulators).

STRICOM is sponsoring the application of structural modeling to ground vehicle simulators.

CECOM is sponsoring the development of domain models for movement control.

NIST is sponsoring the development of domain models for network management systems alarms management.

3.4.5 Product Quality Engineering

This section discusses the product activity area related to product quality engineering.

3.4.5.1 Problem Statement

Software quality requires mature technology to predict and control quality attributes. If the technology is lacking, even a mature organization will have difficulty producing products with predictable performance, dependability, or other quality attributes. Designers often focus on achieving some narrow goal (e.g., performance) while neglecting its impact on other attributes (e.g., dependability). The result is that systems often fail to meet requirements, i.e., they lack quality. Poor quality eventually affects cost and schedule because serious problems are often not discovered until the system integration phase, when remediation would require extensive rework.

The problem arises not just on customized software or software developed under proprietary standards. Open components and subsystems are designed to meet the same interface standard, but different components or subsystems could emphasize different (and perhaps conflicting) quality attributes. Integrating open components or subsystems of different provenance could have an adverse effect on overall system quality.

These attributes are of special interest to the mission-critical computer resources (MCCR) community because there is a lack of:

- A framework for systematic analysis of engineering tradeoffs between product quality attributes.
- Open standards for MCCR systems with stringent real-time requirements.
- An open standards-based software architecture for systems with real-time requirements.
- Timely education for managers and practitioners on open systems and on the state of the art in quality attribute technologies.

3.4.5.2 Customers

Customers for these products are software engineering researchers (technology for quality attributes), developers, and technical managers (tradeoffs between quality attributes).

3.4.5.3 Rational

This product activity area focuses on the prediction and control of those attributes that determine a software product quality and on the use of open system components to build systems with specific quality attributes such as performance, dependability, and interoperability. The

ultimate goal is the ability to quantitatively evaluate and trade off multiple quality attributes to arrive at a better overall system, as suggested in Figure 3-32. To simplify the problem, we will start addressing pair-wise tradeoffs (e.g., performance vs. dependability, performance vs. interoperability). This product activity area pays special attention to the needs of the MCCR community for open systems standards.

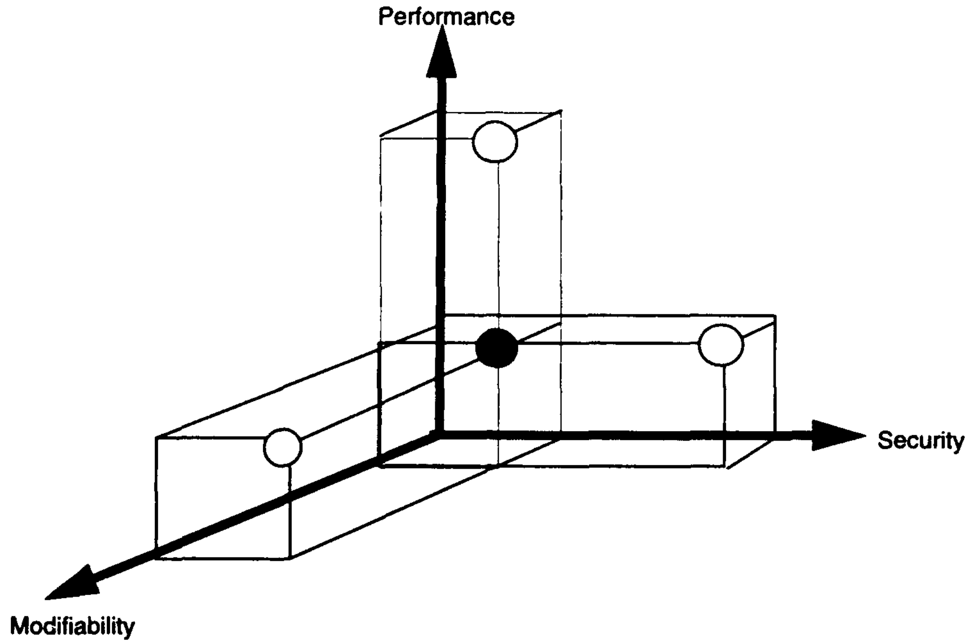


Figure 3-32: Tradeoffs Between Product Quality Attributes

3.4.5.4 Benefits

The benefits from this product activity area can be classified in four categories:

1. **Standards**: with suitable standards for software quality attribute definitions and measurements, and for open system components, the development and maintenance cost and schedules will be reduced.
2. **Architecture**: a "plug-and play" application framework reduces both development and maintenance cost and schedules.
3. **Education**: up-to-date knowledge for managers and practitioners on the benefits and pitfalls of available technology reduces the risk in development and maintenance.
4. **Quality attributes**: technology and prescriptions for practitioners increases the quality and reduces the risk in development and maintenance.

Figure 3-33 describes the trends in product quality engineering over the next four years. Figure 3-34 depicts how this product activity area contributes to the improvement of the software engineering practices in MCCR systems.

Key Items	State of practice as of:			Impact/Metrics
	Today	+2 Years	+4 Years	
Standards	under development	draft standards	accepted	used widely
Architecture	uniprocessor demo	distributed system demo	initial real application	halve the time needed for system upgrade
Education	1st course on open systems and 2nd Dependability workshop	additional courses and workshops	revised and updated courses and workshops	popular course and workshops for managers and practitioners
Quality attribute measurement	subjective; no notion of interactions among quality attributes	feasibility of quantitative measures of qualities pilot use in development programs	handbook for attribute measurement	attribute measurement techniques used by developers
Quality attribute prescription	prescription by heuristics and QA	prescription by architecture	prescription re-enforced by development environment tool sets	organizations using tool sets to generate system artifacts from architectures models

Figure 3-33: Trends in Product Quality Engineering

3.4.5.5 One-Year Objectives for 1995

Analysis of Quality Attributes. The tradeoff the community expects to exercise is between the ability to plug in and plug out components (interoperability) and a nominal loss of other attributes (e.g., performance), but there are often some unwanted results. The work consists of three components:

1. Identify linkages and tradeoffs between dependability and performance to satisfy user requirements.
2. Identify linkages and tradeoffs between interoperability and performance.
3. Trade the degree of fault coverage, the set of permissible online changes, and the performance cost associated with the chosen fault coverage and modification flexibility.

Simplex Architecture for Dependable Real-Time Software. The Simplex[2] architecture is designed to support the evolution of mission-critical systems. It makes the upgrade of computers, networks, and application software easier and safer. It accomplishes these objec-

2. The term "Simplex" is coined from the terms "simple" + "complex." A simple version of a critical component (providing less functionality but more dependability) is used to monitor and, if necessary, override, a complex version of the same component (providing more functionality but less dependability).

tives by encapsulating three advanced technologies—generalized rate monotonic theory, membership protocols, and the theory of analytic redundancy—while giving users a simple "plug-and-play" application framework.

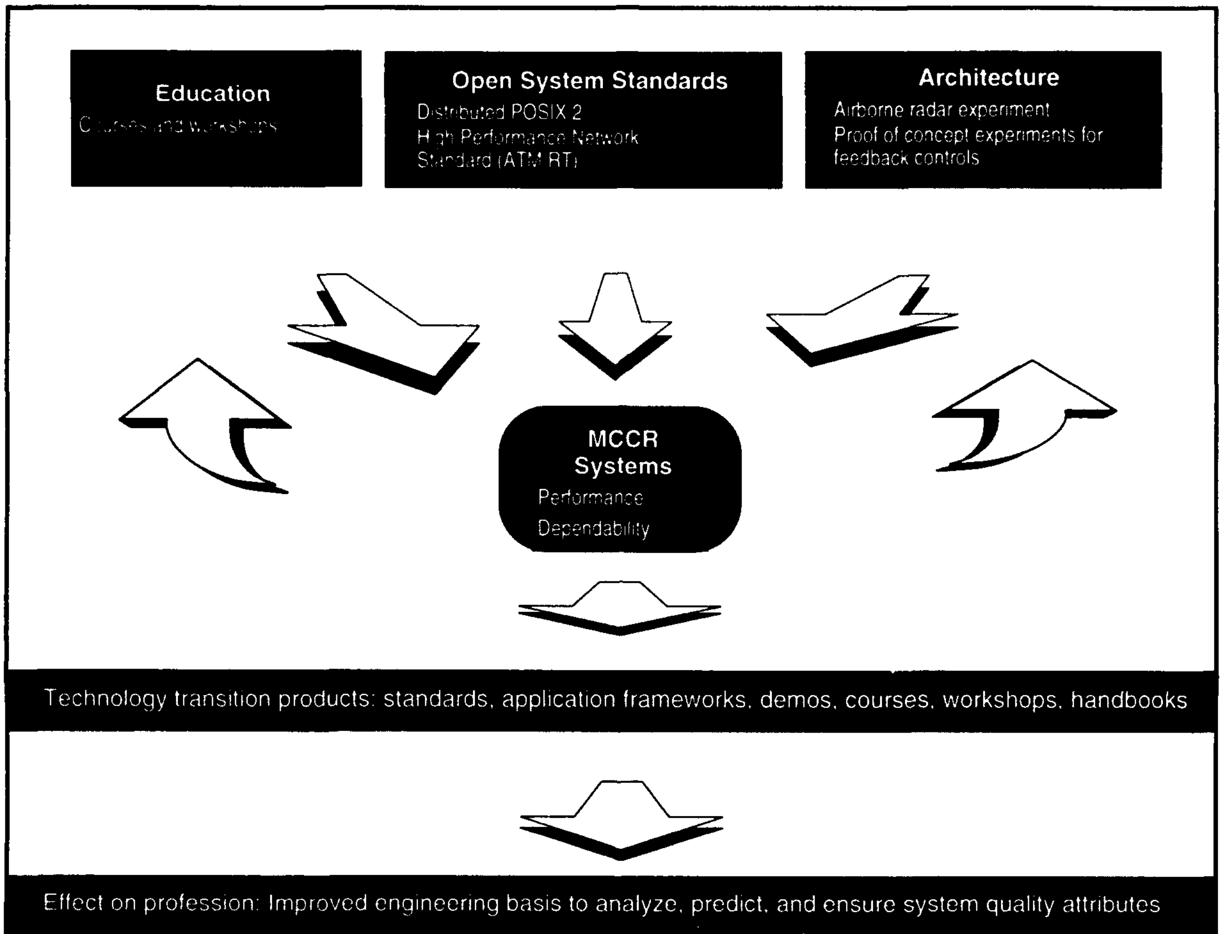


Figure 3-34: Improving the Practice for MCCR Systems

In this work, we will demonstrate the Simplex architecture and transition the principles of this architecture for distributed real-time, fault-tolerant, mission-critical systems through technical reports. Two application areas will be considered: one for mission-critical application and another for the process control of the manufacturing of tactical systems.

Open Systems Standards. The Navy Next Generation Computer Resources (NGCR) Program has been funding the SEI to work on the development of open, dual-use standards that meet the requirements of the mission-critical community. Specifically, there are two tasks:

1. Work with the IEEE 1003 family of standards (POSIX) to develop a standard suitable for the real-time distributed systems communication domain. This work is partially supported

by NGCR. The NGCR funding for POSIX 1003.21 does not cover important subjects such as the formal specification of the POSIX 1003.21 standard.

2. Work with the Asynchronous Transfer Mode (ATM) Forum and the NGCR High Performance Network working group in the development of the Navy's Next Generation High Performance Network Standard. The current candidate is the Real-Time Extension to the ATM (ATM/RT) standard.

The SEI is leveraging its involvement in this standards work and plans to develop a handbook for using open system architectures in mission-critical systems.

The development of open standards cannot be divorced from the development of standards for software quality. The developers of POSIX 1003.21 are paying special attention to the inclusion of quality attribute specifications in the standard. These specifications take the form of schedulability metrics, i.e., factors that determine performance-related properties of an open component and are made part of the component "data sheet." As the technologies for predicting and controlling quality attributes mature, additional quality attribute specifications should be routinely included in open system standards.

Several proposed add-on activities provide additional leverage to the baseline investment. DE-6A and DE-7A extend the work on a dependable software system architecture (Simplex); DE-1A, DE-2A, and DE-8A investigate technology support and collect experimental data on impact of architectural choice on system properties (see Sections 3.4.8.1 through 3.4.8.3).

3.4.5.6 Work Outputs

Report on Quality Attribute Tradeoffs (1995). The report will document a framework for systematic tradeoff of quality attributes based on architectural choices. This framework will be validated through tradeoff studies between fault coverage, modifiability, and performance.

Airborne Radar Study Reports (1995). We have been applying the Simplex architecture in a proof-of-concept demonstration in cooperation with another federally funded research and development center (MITRE) in a simulated environment that models the problems encountered in the Airborne Warning and Control System (AWACS) Program Upgrade. This work is led by MITRE, and the milestones are determined by AWACS program needs. We anticipate that one 1995 output will be a report, based on MITRE and AWACS schedule and requirements.

Open Systems Handbook (1995). We plan to develop a handbook for using open systems architectures in mission-critical systems. This handbook will be architecture-based and will highlight issues and approaches for solutions to the major problems facing developers. It will build on open systems standards work that the SEI has been involved in. The development of standards is a time-consuming process. It often takes years to reach official approval, with multiple intermediate drafts circulated for comments and voting by the technical community. During 1994 drafts of POSIX 1003.21 will appear, and the final standard is expected to be approved in 1995. This is also the target date of the open systems handbook.

Rate Monotonic Analysis (RMA) Users Forum (yearly). The 3rd RMA Users Forum will be held in May or June 1995. This recurring two-day event brings real-time practitioners together to share experiences in using RMA, as well as information on new tools and environments that support real-time development. Previous users forums were sponsored solely by the SEI. The 1995 event will be co-sponsored by the SEI and another organization interested in real-time systems (specific organization yet to be determined). Initial planning began in December 1993, immediately after completion of the 2nd Users Forum, and will continue through 1994 and into 1995.

3.4.5.7 Related TO&P Activities

The Office of Naval Research partially funds the integration and demonstration of new technologies.

NGCR and the Space and Naval Warfare Systems Command (SPAWAR) partially fund the development and specification of open standards.

The Naval Surface Warfare Center, NIST, and Phillips Laboratories partially fund technology transition activities.

3.4.6 Automation of Engineering Practice

This section discusses the product activity area related to automation of engineering practice.

3.4.6.1 Problem Statement

In the development and maintenance of large software-intensive systems, much information must be generated, recorded, and transitioned among the people participating in a project. The automation of these activities increases productivity and reduces human errors. In this product activity area, we concentrate on the adoption and integration of CASE tools, and on the management of software engineering information.

The introduction and use of CASE technology is often chaotic for a variety of reasons such as:

- Unavailable evaluation techniques.
- Non-existent measurement of effective use.
- Poor understanding of environment architectures.

Effective access of software engineering information also presents problems due to complexities in:

- Information from different sources and a range of representations.
- Information overload without proper content-sensitive, just-in-time access capabilities.

3.4.6.2 Customers

A wide range of customers will benefit from progress toward the resolution of these problems. Specific examples include:

- Project members who maintain large, complex application systems and who would like better tools to create, maintain, and view the information generated during a project.
- Those responsible for acquisition and introduction of CASE tools into an organization who would like evaluation and selection techniques, and better adoption processes.
- CASE tool vendors and integrators who would like to enhance their tools and integration strategies to meet the needs of customers.

3.4.6.3 Rationale

The DoD and industrial community have invested large amounts of resources into the automation of various software development and maintenance activities. However, the success rate with CASE has been remarkably low. Recent studies have shown that many products become shelfware and are rarely used.

Investments in CASE environments are high, but the payoff has yet to be clearly demonstrated. In fact, we find that assembly and adoption of an integrated environment lacks a systematic engineering approach. As a result, automated support for many software development and maintenance activities is often perceived as intrusive, resulting in rejection.

A particularly acute problem that many projects experience is that much engineering information is either undocumented or inaccessible to the majority of personnel. Undocumented information includes informally stated requirements, rationale, and engineering tradeoffs and decisions. This information can often have a major impact on the design and implementation of a system, and is in many cases impossible to derive from the code. Systematic capture and access to such information, be it about the system or the way the system has been developed, becomes essential for continued evolution.

The increasing amount of information in various forms requires effective access mechanisms to relevant information, on-demand, just-in-time, in order to overcome information overload. This leads to electronic interactive access taking advantage of advances in commercially available technology.

3.4.6.4 Benefits

The work taking place at the SEI attempts to address these problems. The techniques and approaches that are being pursued will have a number of practical benefits to the software engineering community.

CASE Environments

The practical techniques for assessing the qualities of products and integration strategies for CASE environments provide an approach to the selection and use of CASE technology. Our work will result in improved understanding of the issues in engineering a CASE environment, and greater success with the adoption of CASE environment technology.

In the development of evaluation techniques for CASE environments, we are filling a very important need of our customers. We are frequently asked about which tools to use in different situations. The techniques we are developing will help our customers in relating their particular

needs to the strengths and weaknesses of currently available technology. Figure 3-35 illustrates the approach we are following. The number of organizations that apply our techniques (either directly or indirectly) is a measure of our effectiveness.

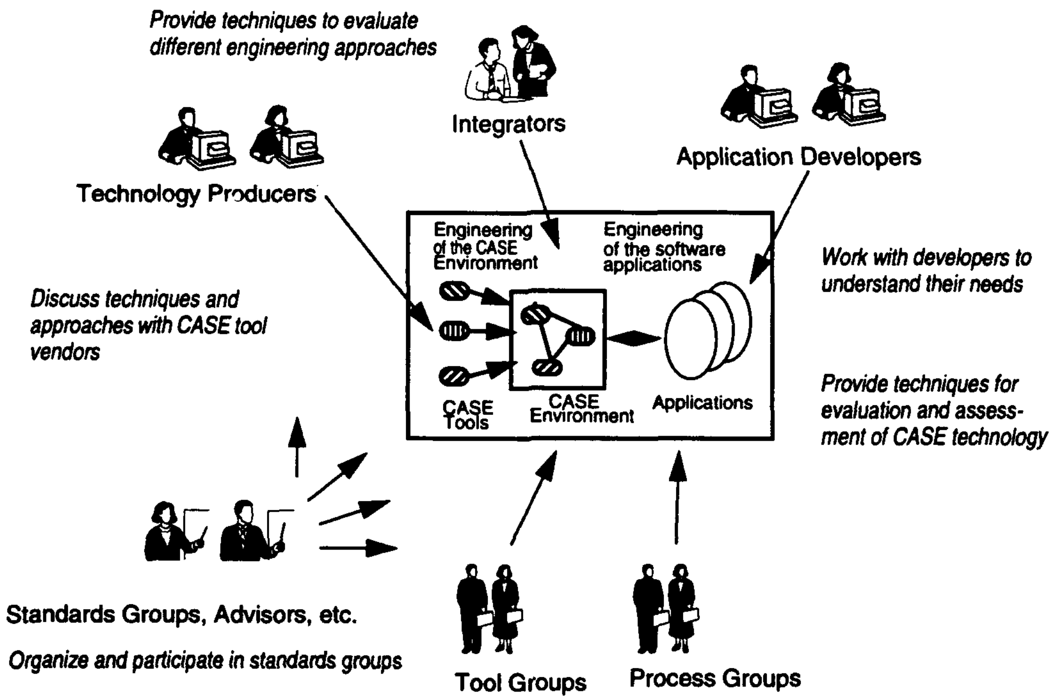


Figure 3-35: CASE Approach

The engineering of a CASE environment from its component parts requires a large number of tradeoffs. Ways to understand, compare, and apply different environment engineering strategies will be of great benefit to the many organizations trying to assemble a CASE environment. We are documenting these different strategies and their effectiveness, and publishing our results widely.

The quality of a technology does not guarantee its successful application in practice. Without practical plans for the adoption of CASE environment technology, almost any solution is likely to fail. We are working to ensure that good adoption practices are readily accessible to organizations that are introducing CASE environment technology.

Figure 3-36 depicts the trends in CASE environments over the next four years.

Key Items	State of practice as of:			Impact/Metrics
	Today	+2 Years	+4 Years	
Evaluation of CASE quality attributes	ad hoc methods	prototype methods; case studies	validated methods; data from users	many organizations using methods
Engineering of CASE environments	no commonality across organizations	best practice guide	updated best practice guide; data from users	publications and guides readily available
Adoption practices of CASE environments	few metrics and no reliable data	defined practices; data from early users	detailed user data; improved practices	standards defined and in regular use

Figure 3-36: Trends in CASE Environments

Effective Access of Software Engineering Information

Our work in this area addresses the following aspects:

- Interactive access to information about software engineering; this can be characterized as an interactive software engineering handbook or on-demand, just-in-time learning. It has the potential of an effective transition medium. The pilot work in this aspect evolves toward collaboration and leverage of outputs throughout the SEI, in particular, education.
- Accessibility of information about a system; system information may be captured in plain text, in various formats and notations, or through interviews with the system architect or maintainer. System design record is a concept that focuses on formalizing relevant information. The pilot work in this aspect demonstrates accessibility to informally captured information in addition to formal designs and implementation. It will be complemented with design record work under the auspices of reengineering and MBSE.
- Evaluation of different technologies in providing effective software engineering information access; a number of commercially available and research technologies can be combined and applied in the context of software engineering. A low-entry cost technology is X-Mosaic and the World Wide Web. Other technologies involve NLP, multimedia, and intelligent tutoring. The SEI is in a good position to leverage an existing testbed at CMU, referred to as Informedia.

We are evaluating the utility of this technology to determine whether we can use it in the delivery of software engineering education and in recording requirements elicitation and review (database query) on an interactive, multimedia, ready-access basis. Toward these ends, the testbed facility is being created through research in the Robotics Institute at CMU and populated with an extremely large engineering database. This database will contain over 1,000 hours of video course material in software engineering, computer science, and potentially other disciplines. The facility will demonstrate intelligent, automatic mechanisms providing full-content search of, and selective retrieval of, engineering information in multiple formats includ-

ing text, graphics, images, audio, and video. This work will lead to a library of information available to the software engineer in a networked, just-in-time fashion, when and where it is needed as illustrated in Figure 3-38. The SEI is collaborating in this effort for its applicability to software engineering.

Figure 3-37 depicts the trends in software engineering information management over the next four years.

Key Items	State of practice as of:			Impact/Metrics
	Today	+2 Years	+4 Years	
Capture and representation of software engineering information and knowledge	text based; difficult; ad hoc	tests of tools & techniques	early adopters using validated techniques	large, growing software engineering information repository, populated with demonstrable cost/benefit
Widely distributable electronic software engineering information products	simple examples; based on outdated design	wide availability of state-of-the-market; evolvable products	wide availability of state-of-the-art electronic software engineering information products	widespread use of repository

Figure 3-37: Trends in Effective Access of Software Engineering Information

3.4.6.5 One-Year Objectives for 1995

- Produce a roadmap for CASE environment technology and continue the expansion and transition of the CASE environment integration testbed.
- Baseline the state of the practice in process-centered environments and expand evaluation techniques and work with customers to pilot and validate them.
- Continue to leverage CMU Informedia testbed work and pilot an Informedia prototype with software engineering information in educational and industrial sites.
- Extend the current X-Mosaic electronic software engineering information base in content, and interface it with the Informedia testbed.

Several proposed add-on activities provide additional leverage to the baseline investment. DE-8A, DE-9A, and DE-10A contribute to systematic evaluation and measurement of SEE technology. DE-1A, DE-4A, DE-6A, DE-9A, and DE-10A contribute to automation and simplification of engineering activities. DE-11A, DE-12A, and DE-14A contribute to advances in maintaining and providing effective access to software engineering information (see Sections 3.4.8.1 through and 3.4.8.4).

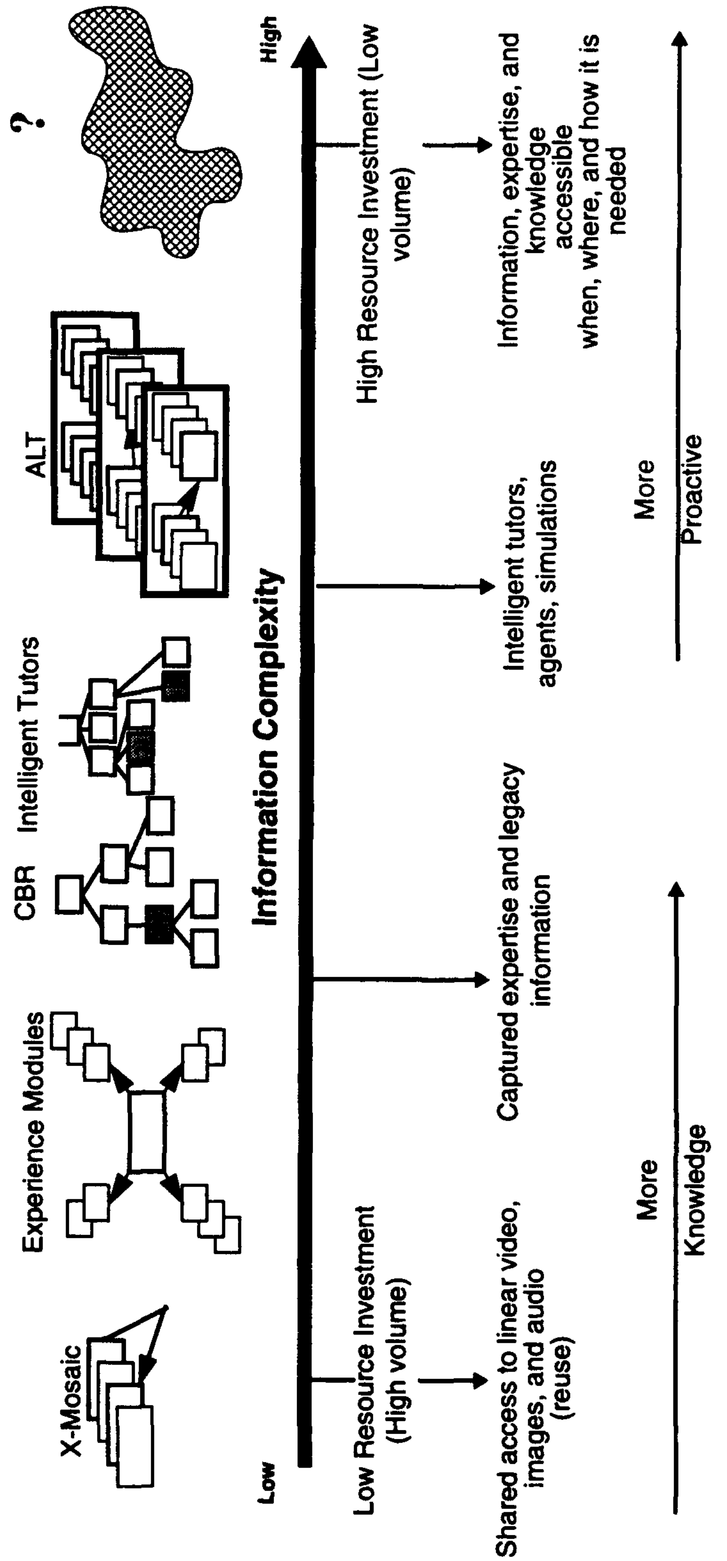


Figure 3-38: Software Engineering Information Modeling Approach

3.4.6.6 Work Outputs

Roadmap for Environment Technology (1995, 1997). The roadmap builds on core investment in 1994 and earlier (*Guide to Best SEE Practice: Edition 1*) and leads to the *Guide to Best Practice in SEEs: Edition 2* in 1997. The work on the 1995 technology roadmap represents a framework for managers and practitioners to understand and relate existing and future environment technology. The roadmap contains such a framework as well as an assessment of maturing and emerging environment technologies. TO&P sponsorship leverages core investment into this technology area. The roadmap will be accessible in the form of a report.

Report on the State of the Practice in Process-Centered Environments (1995, 1997). A technology that has an impact on the quality attribute of fidelity in process automation and requires appropriate methods of assessment. The 1995 report will assess the state of currently emerging, next-generation environment technology with respect to its ease of use for automating software processes. This report will also contribute to the *Guide to Best Practice in SEEs: Edition 2* (1997). There is high interest in this technology by industrial strategic partners, including companies that have signed TCAs.

Electronic Software Engineering Information Base (1995). This output is an operational system providing user-friendly computer-based access to descriptions and outputs of the SEI as well as several pilot examples in selected topic areas of software engineering information accessible for interactive learning. It uses a range of COTS and emerging technologies (X-Mosaic, World Wide Web, Mac, and UNIX-based authoring tools) to maintain and make widely accessible software engineering information. We are developing this system incrementally, both in terms of information content and underlying technology. As a result, an operational information base will be accessible throughout 1995. All product activity areas in this focus area are contributing to the information content. Technology increments demonstrate increased intelligence in information access. They include interfacing to the AMORE multimedia information base prototype (see 1994 Plan) and Informedia testbed, and incorporation of intelligent information indexing and access techniques (see Section 3.4.8.3, DE-11A and DE-12A). The latter part leverages the Informedia testbed work at CMU and has some commitments to the CMU School of Computer Science, the Media and Arts Technology Alliance (MATA), and is of great interest to ARPA, NIST, and the NII initiative.

3.4.6.7 Related TO&P Activities

Integrated CASE, NIST, the National Security Agency, OSD, STARS, and the Deputy Director of Intelligence are providing funding to various aspects of CASE environments objectives. ARPA is providing partial funding for the X-Mosaic based information base. In addition, we are leveraging funding to MATA through our collaboration with the CMU campus.

The SEI has the technical lead in a number of IEEE technical committees and government task forces. In addition, through SEI involvement as technical and editorial lead, SEI work in CASE adoption has become the basis of a draft IEEE/ISO standard, currently in ballot. Similarly, an SEI member is co-chair of the NIST Integrated SEEs forum. SEI members are leading

IEEE committees and are members of government task forces for multimedia technology. SEI members are on the editorial board of Association of Computing Machinery (ACM) and IEEE publications as well as publishing houses specializing in software engineering.

3.4.7 Maturation of Engineering Practice

This section discusses the product activity area related to maturation of engineering practice.

3.4.7.1 Problem Statement

Current software engineering practice is often not documented and systematically used. Baselining current practice, i.e., both technical as well as non-technical issues, is a first step to improving practice. A second step is the evaluation of promising technology as to their maturity for practical use, and their value added to the advancement of practice. The EMM provides one framework for characterizing technology with respect to engineering maturity.

This product activity area focuses on performance engineering and reengineering as perspectives of software engineering from which we address maturation of engineering practice. Other contributors to the maturation of practice are domain engineering, integrated environments, and engineering of flight simulators. They are discussed in the other product activity areas. By performance engineering, we are referring to software timeliness usually measured in terms of latency and/or throughput. Performance problems are very common in software-intensive systems. Yet there is a vast body of knowledge in the performance arena documented in books, papers, and handbooks, including:

- Queuing theory and related performance modeling
- Scheduling theory
- Analysis of algorithms
- Methodologies that account for performance

The fundamental question is: Why do performance problems persist in the face of a highly developed body of knowledge? We are attempting to address and provide answers to this question.

The second area of immediate application is reengineering. As resources become scarcer, the need to modernize existing systems and recover or salvage existing software assets is becoming more critical. This need is accentuated by:

- Rapid technological change
- Trends toward open systems architectures
- Focus on domain models, common architectures, and enterprise models

The SEI is uniquely positioned to exercise leadership in developing a reengineering maturity model and applying these concepts to an understanding and maturation of the areas of performance engineering and reengineering. The SEI is a recognized leader in the areas of pro-

cess, real-time system performance, domain engineering, and environments. This product activity area builds upon and generalizes our expertise in these areas.

3.4.7.2 Customers

Customers for mature performance engineering technology include the broad community of software engineering practitioners and researchers. In addition, customers for the detailed application of reengineering concepts include the reengineering community and PDSS and maintenance organizations.

3.4.7.3 Rationale

The evaluation of the maturity of performance engineering will identify the shortcomings of existing tools and techniques as well as promising techniques that need to be made more practicable. The results of this activity area include identification of tools and techniques with unrealized potential to manage performance and reasons why these tools and techniques have not been more widely used.

These concepts are also applied to the specific case of reengineering because, although a large investment in software-intensive systems is expended on legacy systems, systematic approaches to assessing and evolving legacy (and non-legacy) systems are not in common use. In addition, root causes of undesirable legacy system properties are not easily identified and overcome, while technical as well as managerial and transition issues inhibit effective upgrading of legacy systems. Support for systematic decision making that considers risk, technological, economic, adoption, and managerial issues is not well established in current practice.

Legacy systems require a systematic problem-solving approach using analysis of all information sources to gain system understanding, identification of root causes as well as desired system properties, and determination of an appropriate evolutionary strategy addressing both the evolution of the system architecture and design and upgrading of the operational and support system (see Figure 3-39).

Issues to be addressed in reengineering legacy systems are similar to those of newly developed systems, i.e.:

- Continuous system evolution
- Automation and generation of artifacts
- Design record

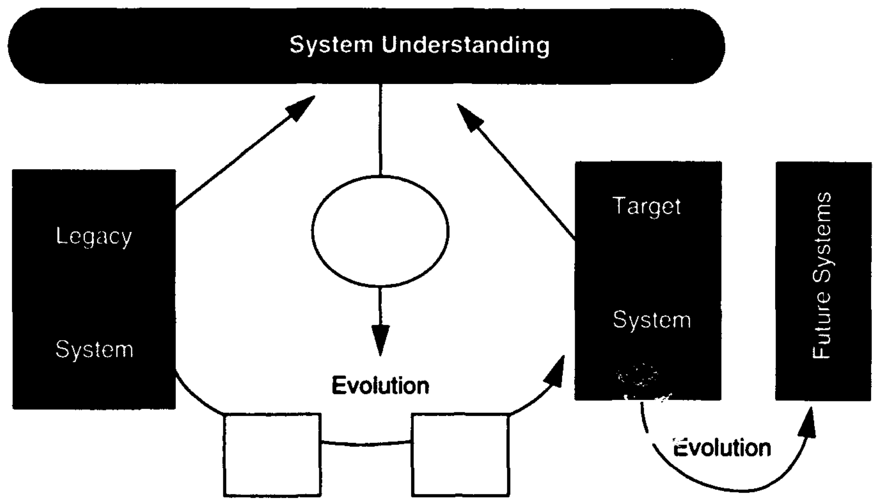


Figure 3-39: Reengineering Problem Solving Approach

3.4.7.4 Benefits

Benefits of maturing performance engineering technology are:

- Improvement in the control of performance attributes evolves from a state in which performance problems are commonly discovered late in development to a state in which mature performance engineering techniques are being used.
- Transition of techniques for controlling software performance evolves from a state in which it is difficult to transition new approaches to a state in which a conceptual infrastructure exists that makes it easier to adopt new techniques.

Figure 3-40 depicts the performance engineering trends over the next four years.

Benefits of maturing reengineering technology include:

- Transition of advances in domains and architectures to the reengineering and maintenance community.
- Acceleration of advances in design record, decision-tree concepts, and other technology developments.
- Development of community consensus.
- A central point of reference for understanding lessons learned.

Key Items	State of practice as of:			Impact/Metrics
	Today	+2 Years	+4 Years	
Methods for identifying high-leverage technical needs, knowledge, and practices in performance engineering	no accepted notion of technical maturity; preliminary interest in notions of knowledge maturity within and outside the SEI	maturity model for performance accepted/used by a few strategic partners and ARPA	maturity model for performance is widely known and accepted; technical maturity is equated to control of quality attributes	number of organization using maturity model for performance; number of product quality success stories attributable to model-inspired improvement
Prediction and control of performance attributes	performance problems commonly discovered late in development	performance engineering rationalized; improvement demonstrated for some strategic partners	control of other attributes demonstrated; performance improvements demonstrated beyond a few strategic partners	number of engineers using mature performance engineering techniques; maturation of other quality attributes
Transition of models and measures of quality attributes	technology transition is generally prolonged, ad hoc, and unpredictable		evidence is obtained that maturity model of performance encourages and accelerates adoption of high maturity practices	transition time

Figure 3-40: Trends in Performance Engineering

Figure 3-41 depicts the trends in reengineering practice over the next four years.

3.4.7.5 One-Year Objectives for 1995

- Develop first version of maturity model for performance engineering technology.
- Identify an important performance engineering problem and performance engineering technologies potentially useful for the problem and start to make them practicable.
- Validate approach for analysis of lessons learned in reengineering.
- Prepare *Guide to Best Practice in Reengineering* and accelerate advances in reengineering technology (design records, decision tree).
- Become recognized as focal point for reengineering efforts and technological innovation and application.

Key Items	State of practice as of:			Impact/Metrics
	Today	+2 Years	+4 Years	
Lessons learned	anecdotal; random	systematic collection and analysis	community repository	common benchmarks avoid reinventing wheels risk reduction
Guide to Best Practice	random; unorganized	common understanding of best practice	systematic updates	transition of best practice to community framework for technology efforts
Design record concept	conceptual identification; not practiced	conceptual refinement; trial use	guided empirical analyses; case studies	potential breakthrough technology; systematic evolutionary development
Advances in reengineering	random	systematic analysis based on architectural descriptions	evolution based on design records	guided advances catalyst for industry technology developments

Figure 3-41: Trends in Reengineering Practice

Several proposed add-on activities provide additional leverage to the baseline investment. DE-7A and DE-13A directly extend the baseline investment in 1995. DE-1A, DE-2A, DE-3A, and DE-7A contribute to an architecture-based approach to performance and reengineering. DE-10A, DE-12A, and DE-14A contribute to advances in reengineering technology, in particular system understanding (see Sections 3.4.8.1 through 3.4.8.4).

3.4.7.6 Work Outputs

Performance Engineering (1995, 1996). During 1993 we conducted a feasibility study on EMMs comparable to the CMM underpinning the process focus area. Starting in 1994 and continuing through 1995, we will conduct a validation study of the EMM in the area of performance engineering. Initial findings of the validation study will be available in reports and presentations. A *Guide to Best Practice for Performance Engineering* will be published in 1996.

Reengineering Practice Guide (1995, 1996). This guide, planned for 1995, will be based on a holistic framework for evolution of legacy systems summarizing best practice, including strategic reengineering approaches, reengineering state of the practice, reengineering decision support, reengineering risk assessment, reengineering method and tool adoption, and reengineering approaches based on domain models and domain-specific architectures. The guide

is being developed both from work done within the SEI and also from broad community input based on the reengineering workshops that have been sponsored by the SEI.

During 1996 we will develop a roadmap for advances in reengineering practice with particular focus on system understanding. This work will build on the *Guide to Best Practice* and will identify promising ideas in reengineering technology such as design records, decision analysis, and architectures. The roadmap will identify the hurdles for further advances and will provide leadership toward the development of an evolutionary strategy to help in the resolution of these problems.

3.4.7.7 Related TO&P Activities

PMA 264 and the DMA are sources of TO&P funding with a focus on reengineering of legacy systems and are candidates for reengineering case studies. Additional candidates for case studies and potential sponsorship include NASA and STARS.

3.4.8 Proposed Add-On Activities

The following section describes proposed core add-on activities in the disciplined engineering focus area. The SEI is asking selected members of the software community to evaluate the relative attractiveness of these add-on proposals as possible elements of the final 1995 core program, as funding permits. The proposals are grouped according to the product activity areas within this focus area. Appendix A lists all baseline and add-on items.

3.4.8.1 Software Architectures and Models

DE-1A Evaluation of Architecture Representation and Analysis Technology

Architecture-based technologies are emerging and turning into potential best practices (CARDS, DSSA, STARS, VCOE, Prototech, etc.). GAO reports have indicated and evidence in industry and government (Ballistic Missile Defense (previously the Strategic Defense Initiative Office, OSD report, HP) shows that a "good" architecture is essential for effective and efficient, high-quality application development. This activity addresses the need for understanding the value-added and maturity of emerging software architecture representation languages and their ability to contribute to architectural analysis resulting in increased predictability of system properties. The SEI has been asked by ARPA to establish a focused effort on this topic and has started this effort in 1994, which as a side effect will promote the transition of architecture-focused ARPA and government programs, including DSSA, ProtoTech, Software Composition, STARS, and CARDS.

This activity has two main components:

1. Establishing a framework for assessment/evaluation of architecture technology supporting the representation and analysis of architectures. The results of this activity are criteria and a method for evaluating software architecture technology (languages, representations, tools). The evaluation method involves systematic hands-on experiments in applying technology instances to real software architecture examples. This activity requires community

participation and buy-in both with respect to the evaluation framework/method and the choice of example architectures for comparison experiments.

2. Validating the evaluation framework through pilot evaluations. Candidates include commercially available solutions (Kaptur); DSSA technology (MetaH, Adage); and software composition research technology (Rapide, UniCon). In addition, this activity will leverage ongoing experiments in the DSSA and Prototech community in describing example architectures in different architectural languages.

This activity lays the foundation for a continuing effort to evaluate emerging architecture technologies (state-of-the-art reports), influence the research direction, and provide roadmaps for reducing the most promising technologies into a continuously improving MBSE practice. Such a repository of knowledge and information is an output for 1996 and beyond.

The activity also lays the foundation for a method allowing practitioners to make choices in the notations, methods, and tools they choose in engineering a software architecture for their system. A 1996 result reflecting this method of choices will be a guide to practical use of architecture representation languages.

DE-2A Case Studies of Software Architectures

This activity focuses on the study of architectures of existing systems and the engineering practices of creating and evolving them. The purpose is to systematically characterize and analyze architectures of existing and past systems for their system properties and relate them to the quality attributes and other benefits found in the operational systems. The result of this activity is an evolving compendium of software architectures of operational systems. Such a compendium may be similar in character to such a compendium for computer architectures developed by Bell & Newell [Bell 71].

The activity in 1995 focuses on a uniform method for architecture comparison based on a small set of canonical representations and descriptions. This method is evolved by drawing on Bell & Newell, on the framework for evaluation of architecture representation technology (see DE-1A), and on research in core architecture description languages. This method will be validated through case studies of several systems, including Ship System 2000 by CelsiusTech and flight simulators. The results will be documented in reports and presentations.

The 1995 activity provides the foundation for a systematic evolution of such a software architecture compendium in 1996 as appropriate. Candidates for this purpose exist in the community, and a number of them are listed in the 1989 GAO report [Paul 89].

The 1995 case studies provide a basis for an activity in 1996 to validate whether predictions of quality attributes based on architecture analysis (see Section 3.4.5.6) relate to actual operational system properties by comparing analysis results with actual system history.

DE-3A Evaluation of a Commercial Application Development Environment Via Prototyping

In 1995 we will develop a selected application system using the SNAP software development environment. This commercial product represents a class of software architecture technology

that can be characterized as an architecture-based application generator. We will evaluate its performance as well as other selected environments should some other strong examples of DSSA-based development environments emerge. The output will be at least one application system and a report of the evaluation of SNAP and its ability to support construction of applications in a domain-specific manner using the DSSA model.

DE-4A System Composition Based on Software Architecture Principles

The work for 1995 will build on results from 1994. Current results include an architectural description language designed to handle a wide range of architectures and an early prototype processor for that language. In the second half of 1994 the prototype tool will be redesigned and rebuilt to allow addition of new component and connector types without major revision of the code.

In 1995 we will extend the architectural model, language, and tool to support a richer set of architectural styles. This will entail expanding the component and connector types (i.e., increasing the variety of building blocks) and developing a classification or taxonomy to show relations among these types and will address a common practical limitation of reuse by finding ways to identify and resolve structural and interface mismatches between components.

The results will be transitioned in three ways: by publications, including technical reports; by demonstration of the prototype tool; and by preparation and dissemination of materials for teaching the concepts and techniques of software architecture.

DE-5A Community Work for Software Architecture Paradigm Shift

This activity supports a number of government and industry programs (e.g., STARS, CARDS, DSSA, Prototech, DoD Reuse Initiative) trying to accomplish the paradigm shift of the software engineering community to architecture-focused software engineering. The activity creates awareness in the community of this coordinated set of initiatives. Three elements contribute to this joint paradigm shift initiative:

1. Development of a shared understanding of a common development and transition strategy by all parties. A first step is dedication of two SEI symposium tracks in 1994 to a coordinated set of presentations. Other steps involve creation of a shared understanding of architecture-related terminology and concepts. The outputs of this component are panel presentations at public forums and a white paper on architectural concepts and terms.
2. Consumer guide to architecture-focused programs. The purpose of this guide is to help customers understand that the results of the different programs and initiatives are not competing technological solutions, but complement each other along both the technology maturity and transition dimension.
3. SEI creation and maintenance of an online information clearinghouse for software architecture-focused results and outputs of different programs and initiatives. This activity builds on the establishment of a software engineering information base (see Section 3.4.6.6).

3.4.8.2 Product Quality Engineering

DE-6A Dependable Real-Time Software System Demonstration

Under the existing architectures for mission-critical systems, it is difficult to introduce new computer or software-intensive technologies into deployed systems. The Simplex architecture is an enabling technology that makes the introduction of new technologies into existing systems safer and easier. This demonstration work is a continuation of the existing 1993 demonstration. In this demonstration, a computer controls two open-loop, unstable devices. The demonstration has shown that it is possible to upgrade the control software online without shutting down the devices. If the new software is better, visibly improved performance will result. If the new software is buggy, the system will maintain the performance of the old software. That is, when the control software is "upgraded," the Simplex architecture guarantees that the system control performance will never be worse than before, despite bugs in the new application software.

This initial prototype was successfully demonstrated at the 1993 SEI Symposium, 1993 IEEE Real-Time System Symposium, and 1993 International Workshop on Responsive Systems. Due to the success of this initial demonstration, we now plan to expand the scope of the Simplex architecture demonstration. We will demonstrate its capability for safe and easy online changes of hardware, network, and application software. Two demonstrations will be developed: one for mission-critical application, and another for the process control of the manufacturing of tactical systems.

DE-7A Dependable Real-Time Software System Handbook

This activity contributes to the development of the technological foundation of the Simplex software architectures, which requires the integration of state-of-the-art technologies in real-time technologies for tolerating random failures (hardware faults) and technologies for tolerating systematic errors (design and implementation faults). We examine a number of challenging technology maturation issues, for example:

- The pros and cons for different technology layering. For example, should the architecture put distributed real-time clock synchronization via datagram at the lowest level or put real-time clock synchronization protocol on top of real-time communication protocols? What are the implications for flexibility, performance, and reliability in different layering decisions?
- In ISIS, a key aspect of its membership protocol is a feature known as the atomic broadcast protocol, which guarantees distributed nodes seeing the identical sequence of events, and thus ensures replication determinacy in distributed event-driven systems. However, atomic broadcast is costly and not compatible with some hard real-time systems constraints. Seeing the same sequence of events too late is not useful since real-time data have a short useful life span. Can we ensure replication determinacy in a time-driven system without atomic broadcast?
- Standard fault-masking techniques such as triplicated redundancy or pair-pair voting requires identical sets of software at each node. However, the use of analytic redundancy for software fault tolerance creates asymmetric software workloads at replicated hardware sites. What should a developer do?

To successfully transition new technologies into practice, promising new technologies often must be matured in such a way that they become compatible with each other and with the existing technological infrastructure. The approach that we will follow is:

- Work with researchers and experienced developers to identify central concepts of system fault tolerance technology.
- Organize and classify the state of the art and state of the practice in fault tolerance technology according to identified concepts.
- Solve the technology maturation problem by working with both researchers and developers.
- Conduct proof-of-concept and proof-of-utility experiments.
- Disseminate the technology with a fault tolerance handbook and a series of technology exchange meetings.

3.4.8.3 Automation of Engineering Practice

DE-8A Software Engineering Environments Technology Evaluation, Integration, and Measurement Initiative

The SEI plays a leading role in the development of a testbed and technology in support of this initiative. This new activity focuses on the state of the practice in SEE technology with particular attention to measurement and evaluation techniques. There is high interest in this initiative (both from government and industry), and we already have a number of highly interested parties with potential co-funding (Air Force Standard Systems Center Gunter, OSD, NIST, and DISA). Interest has also been expressed by Hill Air Force Base Software Technology Support Center (AFB/STSC), Bull, HP, Arcadia, Columbia University, IBM, and Digital.

The activity has two components:

1. Establishing an evaluation and measurement testbed that is shared by the initiative participants and builds on existing work on evaluation and measurement of SEEs in the community. The product of this activity is a testbed that can be used for various kinds of SEE analyses to help the project gain an understanding of the current state of the practice. Elements of this testbed will be transitioned to other government and industry partners involved in the initiative. The testbed will draw on, and be consistent with, the efforts of existing programs such as ICASE and STARS.

2. Developing evaluation and measurement techniques, and validating those techniques. Initially we will concentrate on software bus and object broker technologies such as the Broadcast Message Server, ToolTalk, and implementations of the Common Object Request Broker Architecture (CORBA). These techniques will permit determination of the impact of these technologies, congruence of standards, and evaluation of different integration strategies. The products of this activity will be techniques that can be used by industry and government organizations to benchmark the performance and evaluate the impact of different SEE technologies. We will also produce results from analyses we have carried out using these techniques in a number of key areas such as software bus technology.

As an important part of this work, data reflecting the impact of SEEs will be gathered on specific SEE technology. These data will be used to:

- Estimate the value of SEE capabilities to specific organizations.
- Assess the impact of specific SEEs on an organization's software productivity, quality, and general development and maintenance processes.
- Assist in the identification of process automation practices that have a positive impact on the development and maintenance of software.
- Provide data that can be used to guide current and future SEE procurement in industry and government.

A number of government and industry organizations (e.g., Hughes, Motorola, Digital) have expressed strong interest in these data.

This activity will continue into 1996 with initial commitment for continued matching fund sponsorship from external parties.

DE-9A Assessment of Collaboration Technology

This activity is a feasibility study of technologies for collaboration (also known as CSCW or Groupware) and draws on two emerging themes in the software community. The first is the imminent availability of the NII, which provides the opportunity for the use of SEEs distributed over a wide area and requiring collaborative technology that can operate within multiple and perhaps competing sets of constraints. For the practical success of the NII, applications such as distributed SEEs must be investigated and their strengths and limitations assessed. NIST has expressed a strong interest in this area.

The second emerging theme is the development of techniques for collaborative work in a number of domains. For example, in software development, office automation, and manufacturing system design there is a strong need for technology that can help distributed groups of engineers carry out collaborative tasks. Products are beginning to emerge that can help with these tasks. In software development, for example, tools for collaborative design, inspection, and verification of designs and code are becoming available. Many potential SEI customers are multinational organizations and are struggling with problems of developing software systems in multiple, geographically dispersed locations.

This activity will focus on initial assessment of emerging collaboration technologies to determine their strengths and weaknesses, understand the domains in which they can have positive impact, and identify the major impediments to their success. Examples of technology we will consider include Scrutiny from Bull HN Information Systems, a suite of collaborative inspection and review products for software engineers, and Conversation Builder from the University of Illinois.

The product of this activity will be an assessment of the capabilities of current collaborative technologies, including an analysis of their readiness for full evaluation, pilot use, and introduction into industry and government organizations. This work is being seeded for potential focus in 1996 when more detailed analyses and experimentation can take place.

DE-10A Analysis and Transition of Application Generator Technology
This new activity is an analysis of application generator technology in practice and the transition of this proven technology to new domains. A rich body of such technology, both in the form of application generators and tool generators, has been in use since the 1960s, but is applied effectively only in certain domains. Traditional use of this technology has been for the purpose of automatic generation of application code from high-level descriptions. More recently, generators have been used to produce wrappers and filters to foster interoperability between components that originally were not designed to interact.

One component of the activity focuses on baselining the state of the practice regarding generator technology. This involves a characterization of a range of generator technologies and their scope of application. Technology candidates include parser generators and transformation systems (UNIX Lex and Yacc, Reasoning System Refine, Information System Institute's [ISI] AP5/CLF); 4th Generation Languages; graphical user interface (GUI) generators, wrapper generators (HP Encapsulator, DEC En-CASE, Arcadia Chiron, Marvel wrapper language); interactive tool constructors (Prototech Honeywell MetaDome, Cornell Synthesizer); and algorithm synthesis (Kestrel KIDS). The results will be documented in a report.

The second component focuses on case studies of different application domains for applicability of existing generator technology. The purpose of these case studies is twofold. First, they provide evidence of successful use of generator technology in new domains. Case study candidates include ISI's use of language generator and transformation technology for message validation software in DSSA, and use of wrapper and GUI generators for interactive browsers to object systems. The results will be documented in reports. The second purpose is to identify new domains for pilot use of generator technology. This aspect is accomplished through collaboration with other SEI activities that focus on specific application domains.

A continuation of this activity in 1996 is the pilot application of generator technology in selected new domains, and investigation of inhibitors to rapid transition of this well-known technology to those domains and ways to reduce them.

DE-11A X-Mosaic and the World Wide Web as an Enabler for Innovation Demonstration
The purpose of this activity is to produce a demonstration that exploits the X-Mosaic and World Wide Web technology and its information repositories through intelligent information access. The ability to intelligently navigate large volumes of information should stimulate many new innovative ideas in such diverse areas as software engineering education delivery, publications and document control, intelligent browsing of software artifacts, and browsable reusable software repositories. The result of this activity will be an interface between a more intelligent Informedia testbed and the X-Mosaic information repository representation Hyper Text Mark-

up Language as well as use of software engineering information sources being created through the CMU campus collaboration. This allows us to demonstrate a continuum of technology being applied against large volume software engineering information bases and to perform investigations on the effectiveness of different technologies for on-demand software engineering information access and learning.

DE-12A Effectiveness of Software Engineering Information Base Technologies

The purpose of this activity is to measurably demonstrate the effectiveness of intelligent access to software engineering information bases, both as a transition medium and as a tool for software engineering practitioners. The findings provide tangible evidence of the benefit and effectiveness of the SEI (and other organizations') investment in software engineering information bases ranging from low entry cost solutions such as X-Mosaic and the World Wide Web to intelligent Informedia capabilities (see DE-11A). The results of this activity also contribute empirical evidence to the nation's investment in the NII.

This activity has two components:

1. Evaluate and measure the effectiveness of the software engineering information base as a technology transition medium for the SEI. Empirical evidence of X-Mosaic-based information bases, both in network and CD-ROM formats, provides a set of baseline data to relate to traditional transition media and to demonstrate the value added of intelligent Informedia technology. The findings will be documented in reports, presentations, and in the information base itself.

2. Demonstrate, evaluate, and measure the effectiveness and value-added of intelligent Informedia technology on the existing software engineering information base. Informedia technology increments start with the existing Amore prototype and include technology additions such as NLP and image processing. These technology additions for automatic creation of information hyper-structures are emerging through cooperative work on the CMU campus. The results are improvements in the operational prototype and reports documenting the findings.

Work in this area is of high interest to ARPA, NIST, IEEE, and the NII initiative. SEI members act in an advisory capacity to these bodies in the formulation and refinement of visions, strategies, and programs. The media and communications industry (Microsoft, Time-Warner, Bertelsmann) are strongly interested in this technology as a new medium for providing information and entertainment. SEI Products and Services, SEI Public Relations, and SEI technical focus areas are interested in its effectiveness as a transition medium. Last but not least, SEI members have provided leadership in collaborative efforts known as the MATA with the CMU campus (School of Computer Science (SCS), the Robotics Institute) and others (QED, Intel, Microsoft, and Digital) with respect to synthesis of a range of existing COTS and research technologies for Informedia and digital video libraries.

3.4.8.4 Maturation of Engineering Practice

DE-13A Business Strategies for Model-Based Software Engineering (MBSE)

This activity focuses on business case analysis for software engineering based on software architectures and models. This activity results in a framework for making business decisions of whether a company should engage in architecture-focused technology including domain-specific reuse, product-line engineering, and reengineering. This activity involves collaboration with experts on campus (Graduate School of Industrial Administration, the Heinz School of Public Policy and Management) as well as SEI customers in industry and government who have expressed interest in it (Motorola, US West, HP, Bertelsmann). The results from this activity, together with the technology transition assessment and planning method (see Chapter 4), will provide essential components for bridging the gap between early adopters (trial/pilot use) and the late majority (adoption/institutionalization) (see Section 3.4.4).

The framework for evolving business strategies for MBSE includes techniques for:

- Identifying the domain of core competence and related core assets in an organization.
- Baselining organizational, investment, and development strategies for systematic software engineering based on architectures and models (reuse, product line engineering, reengineering).
- Assessing suitability of systematic reuse and rapid application development based on domain models and architectures for a product group.
- Developing a business case for the above, and techniques for evolving an organizational strategy for transition engineering to an architecture focus in software engineering.

There has been a strong need as well as interest in collaboration expressed for this activity by industry. It is being considered a critical element for successful introduction of domain engineering and application engineering based on a domain model and architecture. The SEI is in a unique position to address this issue as the SEI represents neutral ground for companies, and it can foster interdisciplinary collaboration (including business, industrial administration, and policymaking on the CMU campus as well as at the University of Pittsburgh). An initial framework will evolve and be documented in 1995. The activity is expected to continue into 1996, resulting in a guide to business case analysis for MBSE practice.

DE-14A Prototype Design Record for a Multi-Supplier Software Component Industry

This task develops a prototype design record for instantiation and validation on actual projects. The concept of a design record has been developed at the Microelectronics & Computer Technology Corporation, NAWC, and through ARPA's DSSA Program. While the concept shows substantial promise for improving the evolution of software, it has not been instantiated on actual projects. The task will develop a prototype design record and address the issue of what is relevant information in a design record to foster a multisupplier software component industry.

Concerns regarding such components include openness and interoperability. We intend to approach this task with two initial activities. The first activity focuses on information relevant and

critical to practitioners in PDSS. The emphasis is on information relevant to maintaining and evolving a system.

The second activity focuses on information relevant and essential to support effective multi-supplier management of software components and their integration into a complete system.

The former activity builds on previous work by NAWC. The latter activity will be in collaboration with the CMU SCS and with several industrial partners.

This task is a stepping stone for increased work in 1996 in system understanding and design records. The task has interest from NIST for fostering a software component supplier industry.

3.5 Trustworthy Networks

In November 1988, in response to an automated attack on thousands of Internet-connected computer systems, ARPA established the CERT at the SEI. CERT's specific mission is to:

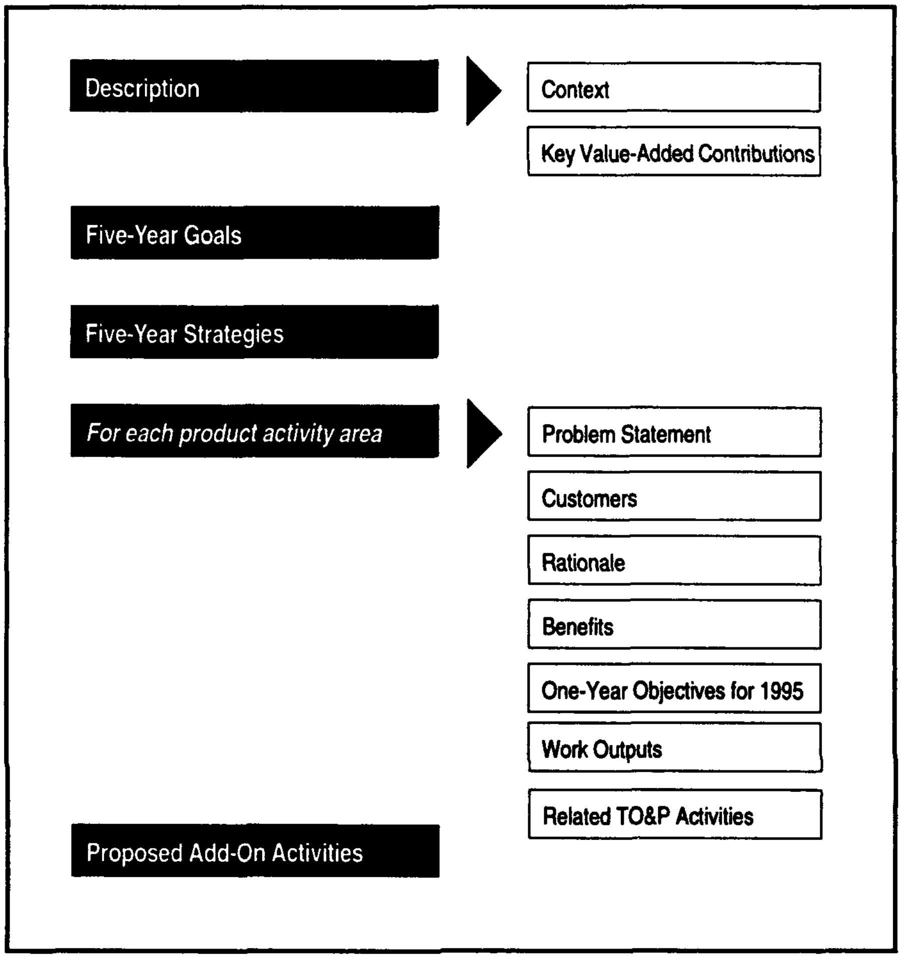
- Provide a 24-hour single point of contact for emergencies.
- Facilitate communication among experts working to solve security problems.
- Serve as a central point for identifying and correcting vulnerabilities in computer systems.
- Maintain close ties with research activities and also conduct research to improve the security of existing systems.
- Initiate proactive measures to increase awareness and understanding of information security and computer security issues.

When CERT was created, the Internet had 80,000 host computers. Since then, the network has grown to more than 2.2 million hosts with an estimated 20 million users. During that same period, the computer security incident rate proportionately increased. The CERT staff has responded to over 2,900 computer security incidents. In addition, it has issued over 100 advisories as a result of widespread intrusions or to warn the Internet community of vulnerabilities being exploited by intruders. These vulnerabilities represent failures in the software engineering practices of technology producers and vendors today.

As the Internet becomes larger and more complex, as the NII evolves, and as larger numbers of organizations become dependent on off-the-shelf technology and open access wide-area networks, the frequency and severity of unauthorized intrusions into systems connected to these networks become increasingly serious. The integrity and availability of the data stored and processed on those networks, and the operation of the networks themselves are at stake. Traditional security measures are not sufficient because of the open access of the networks, the open architectures used to build the networks, and the widespread and distributed nature of the intrusions.

As an outgrowth and planned extension of the CERT activity, the SEI intends to establish trustworthy software as an area of technical focus. Initially we will focus more narrowly on trustworthy networks. The goal is to help the network communities build and maintain trust in wide-area networks by decreasing the risk of computer security incidents. While other system security efforts provide protection through compartmentalization, classification, and extensive evaluation of products, our effort focuses on developing and transitioning information security and computer security practices that are sensitive to the culture and meet the needs of evolving network communities. Because, increasingly, systems are being built by integrating existing software components, we also plan to explore and mature software technologies that allow system integrators to predict the security characteristics of systems built from these components. Finally, to verify system conformance with expected behavior and system operation with organizations' policies, we plan to explore and mature technologies that monitor system configurations and performance.

The sections for this focus area are:



3.5.1 Description

3.5.1.1 Context

From the CERT experience of the past five years, it is evident that the software engineering practices of open-systems technology producers and vendors are not adequate to counter the threat of network and computer system intrusions. The symptoms of the failures are computer security incidents, but the root causes of these incidents can be attributed to the failure to apply sound software engineering, for example:

- Requirements definition practices that fail to account for the need for simplicity in system administration practices.
- Software architectures that provide user required functionality only by sacrificing basic system control principles.
- Errors in design that lead to unexpected system behavior that can be used to gain unauthorized system access.

- Errors in implementation such as weaknesses in coding, review, inspection, integration, and testing practices.
- Weaknesses in configuration management practices that lead delivery of insecure configurations to users.

In short, the rate and severity of computer security incidents serve as a barometer for the state of the practice of software engineering in a large segment of the community.

Our work in trustworthy networks is focused on bringing SEI core competencies to bear on the problem of information security and computer security in wide-area networks. Our planned approach to the problem includes computer security incident response activities that help the network community deal with its immediate problem while allowing us to maintain an ongoing understanding of the problem and the community's needs. The approach also includes near-term and long-term activities focused on preventing incidents. Near-term work aims at providing network and system administrators and users with tools and techniques they can use to assess and improve their network security. Long-term work aims at dealing with the root causes of the problem by exploring, developing, and transitioning improved software engineering practices to technology producers and vendors who supply the wide-area network market.

3.5.1.2 Key Value-Added Contributions

To date, the bulk of our work has been in security incident response and security awareness activities. Much of this work has built relationships and infrastructures that position the SEI to take the next step toward dealing with the root causes of the problem. Key contributions include:

- Development and operation of an incident response capability that has facilitated the resolution of over 2,900 computer security incidents and captured data on those incidents to support the analysis of vulnerabilities as well as the analysis of threats. This analysis has allowed us to issue over 100 advisories to the community. These advisories give system administrators corrections to vulnerabilities or profiles of attack techniques that allow the administrators to monitor for unauthorized behavior.
- Development of working relationships with 32 computer software and system vendors. We work closely with this vendor community to inform them of security deficiencies in their products and to facilitate and track their response to the problems. We work with the vendors to improve the security they design into and deliver with their products. We also encourage them to add security topics to their standard customer training courses. It is our plan to evolve these relationships so that the SEI influence appears earlier in the product development cycle and helps the vendors address the root causes of the problems in their products.
- Playing a lead role in the formation of the Forum of Incident Response and Security Teams (FIRST). Working with private organizations and government agencies we helped them form their own incident response teams and an association that helps these teams work together. Originally called the CERT-System, this collaboration consists of 27 teams that work together on incident response and prevention. The FIRST is a natural distribution channel for the security technologies and techniques we plan to mature.

- Development of two security seminars, sponsorship and organization of ongoing security conferences and workshops, and creation of press relations that help raise awareness of information and computer security issues and of technologies and practices available to deal with those issues.

3.5.2 Five-Year Goals

Our plan is to make the Internet and other wide-area networks more secure by focusing on incident response, improving security technology, and improving security practice. In particular, the goals are to:

- Foster global incident coordination and resolution by facilitating the continued creation and coordination of incident response teams. It is our goal to have, by 1998, an incident response team for each regional network service provider so that routine, limited-scope incidents can be handled at the regional level. In addition, by 2000, our goal is to have an incident response and security infrastructure in place at 50% of major Internet sites.
- By 1997, regional network service providers and major Internet sites will routinely use a set of techniques and tools that enable system administrators to monitor and improve the level of security on their systems and networks. By 2000 these tools and techniques will be available to the broad community through the network service providers and other components of the network infrastructure.
- By 1998, security incidents caused by errors in software architecture design, or implementation will be reduced by 50%. Security considerations will be routinely integrated into requirements and design specifications and, by 2000, will be incorporated into the entire software life cycle.

3.5.3 Five-Year Strategies

Our strategy is to use the incident response activity to benchmark the state of the practice of information system security and to use that knowledge to help the network community improve their products and practices.

We combine our expertise in responding to network intrusions with knowledge of security practice and technology to raise the level of security on the Internet. We use activities that generate information and knowledge, such as incident handling and technology exploration, to drive other product development activities.

To gain leverage with our limited resources, we concentrate on working with other response teams, network service providers, technology producers, vendors and leading network users to transition network security practice and technology to the Internet. In the next five years we plan to develop a large and diverse set of distribution channels.

The trustworthy networks focus area uses an established advisory board to provide advocacy and community feedback on strategy and outputs. See Appendix B for more detail on that board.

Finally, we plan to use the technical results of other SEI work, such as that in the risk and disciplined reengineering focus areas, as the basis for a set of products and services that can be focused on the information system security problem. By transitioning SEI work to a new audience and by repackaging some of it to address new problems, we plan to both broaden the impact of the SEI work as well as extend its scope to a new problem area.

3.5.4 Incident Handling

This section discusses the product activity area related to incident handling.

3.5.4.1 Problem Statement

As the Internet and NII become larger and more complex, the frequency and severity of unauthorized intrusions into systems connected to these networks become a serious problem. The data stored and processed on these networks will be at risk; the need to protect information and resources is critical. In the absence of a centralized response and coordination facility, the resolution of these problems is difficult at best.

Wide-area networks provide an environment where intruders (often referred to as hackers and crackers) form a well-connected community and use network services such as e-mail, netnews, and bulletin boards to quickly distribute information on how to maliciously exploit vulnerabilities in systems. From hobbyists to serious attackers, the intruder community dedicates time to developing programs and sharing information. They have even developed their own publications, and they regularly conduct conferences that deal specifically with tools and techniques for defeating security measures in networked computer systems.

In contrast, the legitimate, often overworked system administrators and users on the network frequently find it difficult to take the time and energy from their normal activities to stay current with this information, much less design patches, workarounds (mediation techniques), tools, policies, and procedures to protect the computer systems for which they are responsible. Moreover, the legitimate network community is not as well coordinated; administrators and users work independently, trying to protect their own systems from harm as best they can.

3.5.4.2 Customers

Customers of the incident handling activity include network service providers, security and system administrators, operations managers, and incident response teams. Individual users also look to us for information and advice.

3.5.4.3 Rationale

Incident handling activities help system administrators and managers at sites affected by incidents deal with issues they have never faced before. In many cases, our work limits the damage by stopping the intruders before site personnel would have otherwise detected and corrected the intrusion.

Since the Internet is doubling in size each year, and since the incident rate is doubling with it, incident response coordination is necessary to help the 40,000 new Internet users who appear each month. Many of these new users are not aware of the threats and are ill-prepared to deal with security incidents when they occur.

In addition, the incident handling work allows us to maintain a first hand understanding of the state of the practice on information and computer security. The work helps us understand the root causes of the problems as well as the effectiveness of tools and techniques aimed at dealing with the problem.

3.5.4.4 Benefits

When a security breach occurs, the incident handling activity helps affected sites recover from the problem and prevent future problems by identifying and correcting problems in their systems and developing system safeguards and security policies. To limit widespread damage, our staff coordinates with other sites affected by the same problem, works with computer vendors to identify and correct deficiencies in their products, and, when an affected site explicitly requests, works with law enforcement and investigative agencies. When new problems are discovered, or widespread attacks develop, we issue advisories alerting the Internet community to the problem and recommending steps to prevent or recover from an attack.

New users of the Internet benefit from the lessons learned in this activity since they are captured and made available to users of the Internet through a public archive of security information and products. The archive includes security tools, a security checklist, all advisories, "tech tidbits," and other documents about security, along with answers to frequently asked questions.

The trend chart for this activity area is presented in Figure 3-42.

Key Items	State of practice as of:			Impact/Metrics
	Today	+2 Years	+4 Years	
Incident response teams active	CERT and 27 other teams active	50% of regional service providers have response capability	All service providers and many major sites have response capability	Time to resolve incidents decreases
Incident response guidelines and policies	Incident response teams follow documented policies and practices	Services providers agree on policies for their activities; guidelines widely available	50% of regional service providers require evidence of policies before connection	Percentage of sites with policies; percentage of users able to state policies
Vulnerability Corrections	Vendors' response uneven; much variance in time to correction	Major vendors' response uniformly fast; increased electronic distribution of corrections	Major vendors have fast channels to their customers	Time from vulnerability report to correction in customer hands

Figure 3-42: Trends in Incident Handling

3.5.4.5 One-Year Objectives for 1995

- Continue our current high level of incident response, and refine procedures to deal with the continuing increase in security incidents and support rapid problem identification, classification, and resolution.
- Expand our Technical Council, a closely linked system of technical experts who help resolve problems when necessary.
- Strengthen working relationships with technology producers and vendors to more effectively advise them of security deficiencies in their products, help them to resolve the problems, and facilitate the distribution of corrections.

3.5.4.6 Work Outputs

Additional Incident Response Guidelines (1995-1996). Incident response guidelines for vendors, technology developers, and system integrators will contain the information needed to work with the incident handling community to eliminate security vulnerabilities in a timely manner. This includes training material so that the vendor, technology developer, or system integrator will be prepared to deal with the rapid resolution of a security vulnerability during a possible emergency response situation.

Additional CERT Advisories (1995-1996). A CERT advisory is a communication to the network community alerting them of vital security information. An advisory may alert the reader to a specific set of ongoing activities that are occurring within wide-area networks, or it may describe a vulnerability that is being exploited and provide corrective actions sites need to take. In developing advisories, our staff works with vendors of the exploited technology and coordinates response to the problem with other security experts. The advisory is widely distributed through mailing lists and newsgroups, and copies of all advisories are available by anonymous file transfer protocol.

Vendor Workshop (1995). The Vendor Workshop will provide a forum where technology producers (vendors) come together to address important security issues. A primary goal of the workshop is to apprise vendors of emerging threats, as identified through our handling of security events, and to help stimulate the move to improved security in commercial products.

3.5.4.7 Related TO&P Activities

The incident handling product activity area is funded entirely by a TO&P with the ARPA Computer Systems Technology Office.

3.5.5 Incident Prevention

This section discusses the product activity area related to incident prevention.

3.5.5.1 Problem Statement

As more users join the Internet, and eventually the NII, and more services become available, the number of intruders and methods of intrusion also increase. To stem the increases in in-

trusion and raise the level of security of networks, we must address the general conditions that allow emergencies to occur and take proactive steps to improve the state of the practice of networked systems security. Both technical and administrative issues must be addressed.

While the computer security research community has, for a considerable time, focused on many areas related to network and multi-level security, there has been only limited concrete technology exploration related to computer security incident handling and practical security on the expanding Internet. Predicting the security characteristics of systems constructed by assembling reusable components is not possible, yet this is exactly what is widely attempted in open system environments. Although technology exits to construct highly secure systems, these systems are often limited in scope and are not economically or culturally acceptable replacements for today's open systems.

Moreover, the Internet is now being used far differently from just a few years ago. As the nature and expectations of users of the Internet change, so does the nature of the services being offered. Where once the architecture and topology of the network had to be understood in detail, the new distributed network services require little, if any, understanding of architecture or topology of the underlying network. Techniques such as firewalls that were adopted as recently as three years ago and that are effective today, are at risk of giving way as the demand for new services changes the very architecture of the network.

3.5.5.2 Customers

Customers of the incident prevention activity include security and system administrators, system managers, network service providers, members of incident response teams, law enforcement personnel, technology producers, and vendors. Ultimate beneficiaries are all users of networked systems.

3.5.5.3 Rationale

For the community to have confidence in networked systems and to improve the state of security, proactive activity is required in three areas: training to raise awareness of security issues and build skills in using tools and techniques that improve security; development and transition of tools and processes that can be used to improve security; and exploration of technology that can be used to meet the challenge of maintaining security in a constantly changing network environment.

3.5.5.4 Benefits

Our planned training activities increase the Internet community's awareness of information security and computer security issues. By drawing on our incident response experience, we are able to produce seminars and courses that are both pragmatic and relevant to customers. Customers receive state-of-the-practice knowledge and guidelines for applying them to their situation. More importantly, increased awareness will lead customers to expect products with improved security characteristics. This change in customer attitude is necessary to provide

technology producers and vendors with the incentive they need to invest in improving the security attributes of their products.

As awareness increases, there will also be an increased demand for tools and processes that can be used to improve the security of existing systems. Since the systems used for our incident handling work are often attacked, we have an excellent source of requirements and a testbed for new technology for testing and enhancing security in existing systems. The tools and techniques that we will provide to our customers have already been tested in practice in the complex and high-pressure incident response activities. By distributing our tools and techniques widely, we help to raise the level of network security without undue investment of time and effort on the part of individual sites.

By anticipating problems and exploring solutions, we can promote good security practice and the use of effective technology as the Internet continues to expand. The ultimate benefit is an increase in the network community's confidence in the Internet, and thus the NII.

The trend chart for this product activity area is presented in Figure 3-43.

Key Items	State of practice as of:			Impact/Metrics
	Today	+2 Years	+4 Years	
Information and systems security policies and practices	Seldom exist	Model policies and descriptions of best practice widely available	Major sites have implemented policy and policy awareness programs	Percentage of users able to state policies
Tools to enhance and monitor system security	Small number of public domain enhancement tools; no monitoring tools	Enhancement tools available from vendors; prototype monitoring tools in test	Enhancement tools in use at major sites	Percentage sites with tools in use
Commercially available products exhibit higher levels of security	Little attention to security; often implemented as post-design patches	Security model and metrics available to guide design activities	Early use of key practices that support measurable improvement in security	Security metrics

Figure 3-43: Trends in Incident Prevention

3.5.5.5 One-Year Objectives for 1995

- Develop information packages that raise the community's level of understanding of technologies and administrative practices that improve security.
- Collect and distribute public domain tools that allow system administrators to test for the presence of known vulnerabilities and security weaknesses in their system configurations.

- Distribute network monitoring tools that allow system administrators to verify that their systems are being operated in accordance with their security policy and to detect anomalous behavior.
- Work with network service providers to broaden information and tool distribution mechanisms for disseminating information broadly.
- Conduct the second annual management conference on information and system security.

3.5.5.6 Work Outputs

Manager's Course (1995). The Manager's Course will be designed to help managers understand what needs to be done to ensure that their computer systems and networks are as securely managed as possible when operating within a wide-area networking environment. Course participants will be provided with information to help them understand and recognize computer security threats, and to make them aware of legal issues surrounding the use and abuse of information technology resources. They will learn how to formulate realistic security policies, procedures, and programs specific to their operating environment, including planning for and management of information technology security incidents.

System Administrator's Course (1995). The main objective of this course is to train system and network administrators in security practices. Course participants will learn how to configure systems and network services in a secure manner, and how to use systems administration tools and techniques to maintain the security of their systems and networks. Students of this course will learn how to actively check their systems and networks for signs of intrusions and other suspicious events. Finally, course participants will be trained in preparing for and responding to security incidents.

Security Checklists (1995-1996). Checklists for vendors, technology developers, and system integrators will be derived from CERT's vulnerability and incident handling data, and will provide guidelines to help prevent the recurrence of common security flaws introduced by poor software engineering and integration practices, by poor choices for default system configurations, and by undue administrative complexity.

CERT Report (1995-1996). The CERT Report will be a compilation of best practice regarding the security of networked information technology resources. The objective of the report is to help management understand both the risks and benefits of being connected to worldwide resources like the Internet. The report will provide an objective basis for comparing practices within their organizations with "widely accepted best practice" as a benchmark of due care in managing risk to their organization's information assets. As a statement of best practice, the report will help an organization set security goals and objectives as well as the policies and procedures that support those goals and objectives.

Network Management Tool (1995). We plan a development effort that will result in a prototype network management tool that can be used by local area network managers to help them detect security problems. Our study of current products on the market and technology development efforts points to a lack of tools for comprehensive local area network management. It

is our plan to prototype such a tool using a unique approach to network activity logging and reporting. It is planned that this tool will provide a comprehensive network transaction audit trail for a complete wide-area network segment. This tool can be used to verify access control policy enforcement, and will provide real-time notification of network security events. Used over time, the tool will provide data on long-term network use trends.

3.5.5.7 Related TO&P Activities

The incident prevention product activity area is funded entirely by a TO&P with the ARPA Computer Systems Technology Office.

3.5.6 Proposed Add-On Activities

The following section describes proposed core add-on activities in the trustworthy networks focus area. The SEI is asking selected members of the software community to evaluate the relative attractiveness of these add-on proposals as possible elements of the final 1995 core program, as funding permits. The proposals are grouped according to the product activity areas within this focus area. Appendix A lists all baseline and add-on items.

3.5.6.1 Incident Prevention

TN-1A Improving Software Design by Adding Security Engineering Principles

Products resulting from this activity will substantially improve the practice of software design by identifying and incorporating the essential elements of security engineering into the attribute engineering work that is currently underway at the SEI. The primary customers are software designers and their managers.

We propose a program of work that begins with an analysis of the CERT's six years of empirical data on software vulnerabilities and methods of attack on networked systems. The results of this work will include the development of security metrics, composite software quality metrics, key practices, and an understanding of security tradeoffs at the point of design instead of as post-design patches. The resulting products are:

- **Initial Security Model (1 year).** We will develop a prototype security model based on a set of security engineering attributes and principles. These will be derived from analysis of CERT's extensive collection of vulnerability and incident data, and from information collected from consultation with internal and external experts. From our analysis, we will also create initial security metrics and a preliminary set of key practices to support measurable improvement in the security of software as compared to the state of the practice.
- **Mature Security Model (1.5 years).** We will further develop and enhance the security model, security metrics, and key practices and insure that the model is updated to reflect changes in the threats identified by ongoing incident activity. This will lead to a composite security/attribute engineering methodology.

- **Transition Products (1.5 years).** Transition of our products to software engineering practitioners will begin with publications and conference presentations during prototype development. As the products mature, we will move to more sophisticated transition vehicles such as workshops (or other training), pilot tests with software vendors, work with standards bodies, and use of transition partners. We will also develop benchmarking tools.

TN-2A Security Improvement Plan for Software Engineering Environments

This product focuses on making a substantial improvement to the state of the practice of software engineering by identifying the security profile of the environment and making strategic changes to address identified areas of deficiency. This work builds on the SEI risk technology, applying its concepts to the networked information systems security domain.

Work has begun in 1994 to transition the risk assessment methodology to the networked computer security domain. Our pilot effort will produce an initial taxonomy and questionnaire focused on identifying the security issues and concerns as perceived by the management and staff of an organization, and an initial set of guidelines for organizations to use in addressing and correcting deficiencies as identified by the security profile.

In 1995, we will develop and pilot a comprehensive security improvement plan. Products will include an enhanced profiling tool that integrates both the qualitative data (identified by the tool developed in 1994) and measurement data that represents the actual environment. This integrated data will result in a more complete and meaningful picture of the security posture of the environment. We will also develop an enhanced set of guidelines for the highest priority areas, thus enabling organizations to act upon the results of their profiles. Using our profiling tool, we will collect data on the state of the practice in various industry segments and will publish the data in technical reports and other publications. We expect this data to define current standards of due care, which individual organizations can use as a basis of comparison with their own security profile.

In 1996, we will mature the profiling tool into a robust tool and refine the overall security improvement approach. This work will include further developing improvement guidelines and launching a training program.

In 1997, the comprehensive security improvement plan will be finished through the completion of guidelines supporting all the areas covered by the profiling tool. We will refine the training program and begin to transfer training responsibility to a third party. We will continue to collect and publish current state-of-the-practice data as ongoing work.

Chapter 4: Table of Contents

4 Technology Transition

The SEI collaborates with leading edge firms to analyze, codify, demonstrate, and disseminate technology transition methods and practice used on both the "push" and "pull" sides of technology transition. We set a standard of excellence for software engineering education and curriculum design. Through direct services, we help organizations initiate and maintain improvements in software development and maintenance practices. Finally, we work with the software engineering community to build the infrastructure necessary for self-sustaining improvement and transition.

Chapter 4 discusses each of these areas of expertise, as shown here:

Because effective practice in the transition of technology lies in the critical path to improvement of the state of the practice in software engineering, we work to:

- Analyze research and best practice in technology transition (both within the software community and in other domains).
- Codify and demonstrate both innovative and traditional methods and practice of software technology transition.
- Disseminate tools, methods, and practices that will increase the effectiveness of technology transition within the software engineering community.

Section 4.1 describes planned work in this area.

Education plays a significant role in technology transition by providing knowledge and skills to practitioners and their managers. Software Engineering Institute (SEI) education-related activities improve software engineering practices and advance software engineering as a profession. See Section 4.2 for more on the role of education in technology transition.

Another key transition activity is providing guidance and advice on process improvement and technology transition for specific organizations. We work with organizations that are influential

leaders in the software community, helping them build and sustain continuous improvement of their software processes and their processes for adopting new technologies. Section 4.3 describes these services of the SEI, demonstrating how we can help to develop an infrastructure where one is needed, open new channels for technology transition, and prepare organizations for the changes that accompany their transition and improvement activities.

The SEI also has a range of other relationships that give our customers opportunities to become involved with us to participate in technology transition activities—including opportunities to influence the development of our products and services. Section 4.4 contains details about how we involve our customers and keep them informed about our work.

4.1 Methods and Practice of Software Technology Transition

Software technology transition occurs throughout technology development from the birth of a technology until its retirement. As shown in Figure 4-1, software technology that has been commercially developed and is in use in an organization has most likely been transitioned at least twice, *between* communities respectively concerned with research and development (R&D), new product development, and adoption and implementation. (In addition, the technology is transitioned as it progresses through its life cycle *within* each community.) Traditionally, these communities have only limited interaction with each other. However, by looking at software technology transition across these established perspectives, we can:

- Identify key leverage points, barriers, and issues in the maturation and transition of software.
- Consolidate and build on lessons learned in software technology transition using a common vocabulary and framework.
- Adapt, develop, and demonstrate effective methods and practices for software technology transition for software technology researchers, product developers, and adopters.

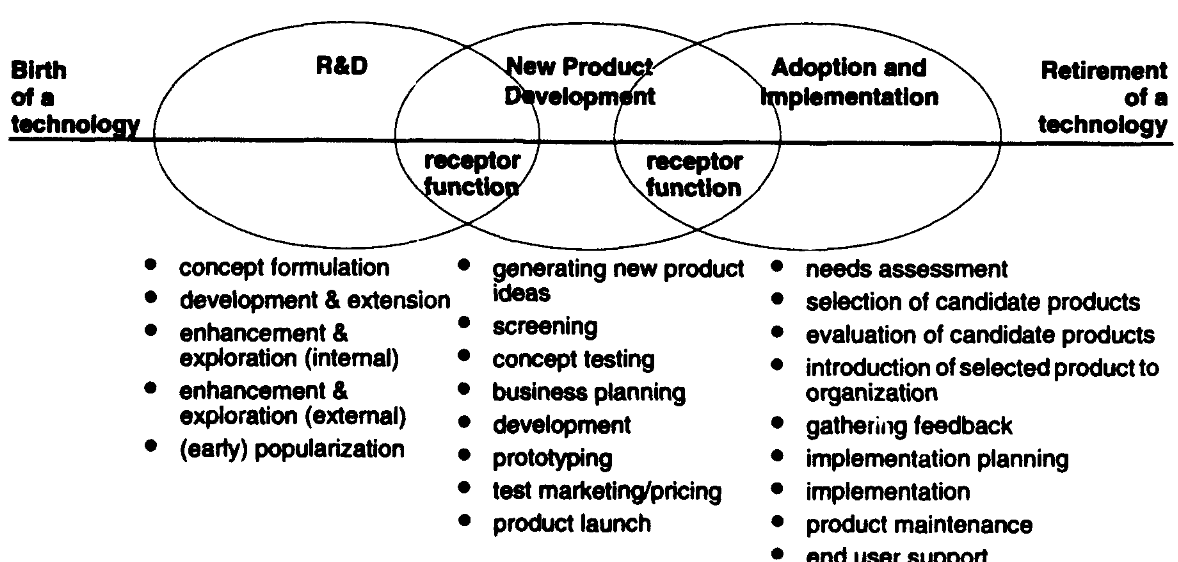


Figure 4-1: The Three Domains of Technology Transition

Common understanding is particularly important to the development of practical approaches to technology transition for the SEI and for its constituencies. Both SEI personnel and SEI customers must have knowledge and skills in technology transition, based on best practice and the application of proven models, approaches, and methods. In the subsection that follows (and continuing in Section 4.1.2), we describe the work we will do in 1995 to improve the practice of software engineering technology transition. By upgrading the software community's capability in this area, so that technology is used sooner and more effectively, we leverage the role of the SEI and increase its impact. This work builds on:

- Earlier foundational work, including the conceptual framework for software technology transition completed in 1993 [Fowler 93a], an in-depth case study of rate monotonic analysis (RMA) from the "push" perspective [Fowler 93b], and a forthcoming case study of RMA from the "pull" perspective.
- Extensive experience transitioning software technology from the SEI, including the RMA handbook and technology-licensing strategies.
- Innovative software technology transition approaches drawn from the software engineering community, such as the adoption services of CaseWare, Inc. and the Hewlett-Packard strategies for "marketing" internal process improvement products.

The *Guide to Addressing Software Technology Transition Barriers* will be developed for use by the SEI and its customers; in particular, it is targeted to those interested in transitioning immature or prototype software technologies or software products, and to those adopting non-process or key process area (KPA)-specific software technologies at Level 3 and above of the capability maturity model (CMM). This guide will be designed to help those involved in both technology "push" and technology "pull" to recognize common barriers to successful transition and to develop strategies for overcoming them. Important work designed to extend and complement this guide, to aggressively advance the state of technology transition practice within the software engineering community as a whole, is described in Section 4.1.2. Key strategies for addressing barriers will be prototyped by the SEI as part of transitioning SEI-developed technology. Based on the results of the prototype efforts, the SEI will define and document a process for technology transition push; this will be refined and continually improved at the SEI and in work with customers from the software R&D and software product development communities.

4.1.1 Work on Methods and Practice of Software Technology Transition

4.1.1.1 Problem Statement

Improvement in the state of software engineering practice requires effective use of both new and existing software technologies and products such as CASE tools, yet these often lack appropriate and effective transition processes. The R&D personnel who develop software technology often do so without awareness of what can make it easier to introduce and incorporate technology into the work of an organization. Change agents, such as software engineering process groups (SEPGs) and advanced technology groups within adopting organizations may

not have the knowledge, skills, and tools needed to effectively introduce and incorporate new or unfamiliar software technologies. The lack of a common framework and vocabulary for software technology transition exacerbates these problems; the lack of all but the most primitive tools (e.g., checklists) constrains even those who are beginning to develop expertise.

Consortia, federal labs, professional societies, government, and other organizations have proposed many approaches to technology transition. These approaches reveal a variety of definitions and processes applicable to software technology transition. When these definitions and processes are taken individually, they provide important perspective on many aspects of transition, but when applied they often result in incomplete or fragmented implementation. Taken together they may appear to conflict, and thus confuse.

The SEI has been successful in transitioning software technology, most notably in the instances of the CMM, RMA, and the master's degree in software engineering, but we continue to seek improvement. Foundaticnal work to develop a conceptual framework and a set of models for software technology transition has been completed and has been used to prepare technology transition case studies and to directly assist customers. Preliminary work in gathering and integrating these approaches is being presented in a tutorial. The next step is to extend and transform the framework and models into products that can be used by both the SEI and its customers independent of direct support.

4.1.1.2 Customers

Customers include all those who can benefit from a more predictable and systematic approach to technology "push" or "pull." These include, on the "push" side, software technology researchers and their sponsors and new software product developers such as tool vendors; on the "pull" side, software developers, purchasers of custom software systems, acquisition managers, advanced technology receptor groups, and SEPGs (especially those adopting and introducing non-process software technologies or technologies at CMM Level 3 and above). Intermediaries—those who facilitate transactions between "push" and "pull" sides—include consultants, trainers and educators, marketers, incubators, and others developing new businesses. SEI personnel are active in many of these roles and thus can act as surrogates for customers during early pilots of innovative software technology transition methods and practices.

4.1.1.3 Rationale

The transformation of a technology as it moves from birth in R&D through product development and into widespread use in target markets is a long, difficult, and typically ad hoc process. As an example, it takes more than 40 years to reach the 10% market penetration level of new manufacturing technology in the United States.[1] Corresponding penetration in Japan

[1] Mr. Ted Olsen, Sr. Vice President, National Center for Manufacturing Sciences, in a presentation to the Technology Transfer Committee of the Council of Consortia in January, 1993.

takes only about 15 years. As an advocate for the software engineering community, the SEI must reduce the time for the deployment of U.S. software technology, as shown in Figure 4-2.

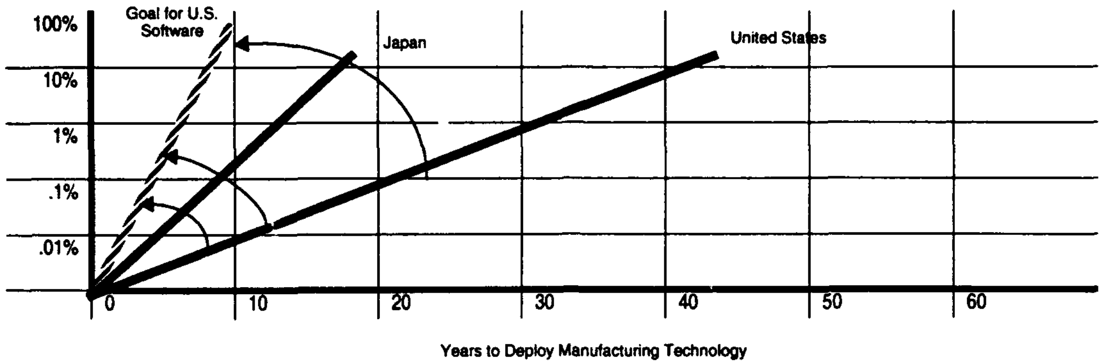


Figure 4-2: SEI Goal of Dramatic Reduction in Deployment Time of Software Technology

Effective software technology transition is on the critical path to improved software engineering practice. The process of introducing software technology can and should be improved—the lack of predictability in the adoption and implementation of new software technology is a cause of software projects being late and over budget. Attention to software technology transition in R&D (especially during prototyping and first-user evaluation) and product development processes can be increased as well and can improve the likelihood of successful introduction.

4.1.1.4 Benefits

Software technology and products built with attention to transition are more likely to be successfully used. Change agents using software technology selection criteria based on "transitionability" are more likely to select technologies that can be effectively and efficiently put to use. Effective software technology transition reduces risk, and to be effective, it should be based on the use of models and best practices proven successful not only in software engineering but also in other domains and in leading organizations. Successful practice must be codified and translated into methods and tools.

Improving our capability to successfully transition software technology will accelerate improvements in the state of the practice of software engineering. A strong capability in software technology transition practice in the software engineering community also:

- Boosts our national competitive advantage by reducing the time for the deployment of U.S. software technology.
- Leverages the investment in SEI work in software engineering technology.

4.1.1.5 One-Year Objectives for 1995

Our objectives for 1995 are to:

- Identify the most common barriers to software technology transition.
- Develop and/or document strategies for addressing these barriers by drawing on existing models and best practices in the transition of technology in general and of software in particular.

Why does popularization and effective use of new technology take so long? The SEI experience in software process improvement suggests that our technology transition process should be repeatable, defined, managed, and ultimately optimizing. As long as the software engineering community uses ad hoc methods for software technology transition, results will be a matter of luck and the very hard work of talented people.

The first step in identifying these barriers is to analyze the software technology transition process from three high-level perspectives. The first, the *life-cycles perspective* already presented in Figure 4-1, allows us to locate the source of the technology being transitioned as well as the ultimate goal of the transition process. This framework is presented in CMU/SEI-93-TR-31 [Fowler 93a], and it makes explicit communication barriers across and within the three major domains of technology transition: R&D, new product development, and adoption and implementation in organizations. The second, the *transition situation perspective,* presented in Figure 4-3, helps us identify the key issues in every transition situation. The analysis of such a situation must consider:

- How the behavior of individual people must change.
- How the organization's structure, process, and reward system must change.
- The characteristics of the technology from a transition perspective (such as whether it must be introduced to all organization members simultaneously or to one unit at a time, incrementally).

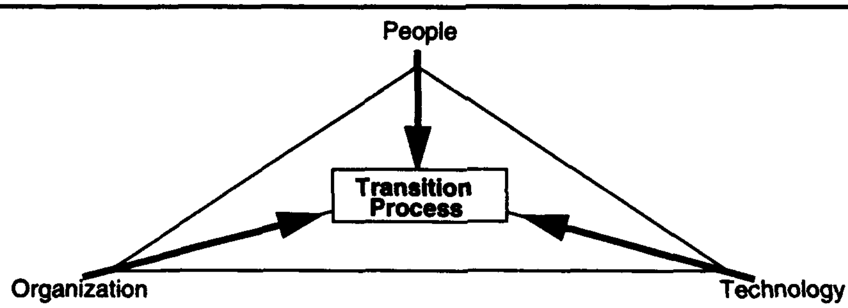


Figure 4-3: Dimensions of Transition

General barriers deriving from the dimensions represented in Figure 4-3 are addressed in an SEI tutorial, *Managing Software Technology Transition as a Project.* A major transition barrier within the organizational dimension is that of organizational readiness for adopting a new technology. Here we propose to add the A VICTORY (Ability, Values, Information, Circumstances,

Timing, Obligation, Resistances, Yield) model [Davis 71] [Davis 79] to the set of models already used.

In the third perspective, illustrated in Figure 4-4 and described below, we view technology transition in terms of changes in the form of the technology as it matures during its life cycle—form-to-form change, or *transformation*.

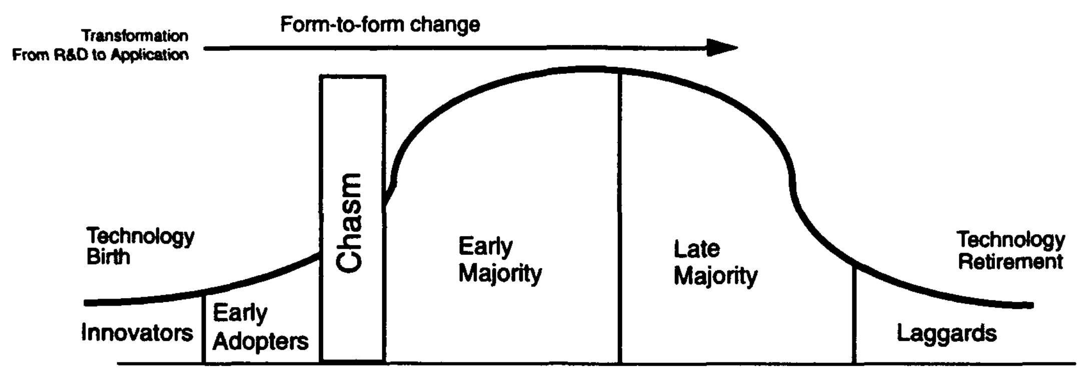


Figure 4-4: View of the Transformation Process

Figure 4-4 illustrates the nature of transformation, showing how the form of a technology changes as different artifacts are created to address different audiences.[2] Form change is particularly dramatic as the technology crosses Geoffrey Moore's "chasm" [Moore 91] and is transformed into "whole" products. RMA technology, for example, has been captured in three different forms since its birth:

- Theorems and proofs in academic papers for innovators.
- Analytical methods initially in experimental field application with early adopters, and later in an engineering handbook for early and late majority populations.
- Tools that are emerging to also serve the majority populations.

Barriers here relate to the difficulty of transforming a software technology to suit these different populations without loss of effectiveness; they also relate to the need to provide adjunct products and services to improve ease of use and of transition.

Once the barriers have been identified and described, they will be translated for the context of software engineering technology transition. A set of questions will be prepared that can be used to identify the barriers within a given transition situation. These questions will be used in planning the transition of selected technologies that SEI projects work on. Based on results from this early application, the questions will be revised and used again in planning the tran-

2. These audiences come from Everett Rogers' adopter population categories [Rogers 83].

sition of selected technologies from other sources. Finally, the *Guide to Addressing Software Technology Transition Barriers* will be prepared and published.

4.1.1.6 Work Outputs

Guide to Addressing Software Technology Transition Barriers (1995). Our understanding of software technology transition as a life cycle, as a transition situation, and as transformation, provides the basis for our development of a *Guide to Addressing Software Technology Transition Barriers*. This guide will provide a catalogue of common barriers and typical strategies for addressing these, drawn from a sample of practices in leading firms. The guide will also provide a set of questions and checklists to help its readers diagnose and characterize particular transition situations—for example, when a CASE tool vendor is planning to provide adoption services to customers.

Guide To Methods and Practices for Software Technology Transition Push (1995). Key strategies for addressing barriers will be prototyped by the SEI as part of transitioning SEI-developed technology. Based on the results of the prototype efforts, the SEI will define and document a process for technology transition push; this will be refined and continually improved at the SEI and in work with customers from the software R&D and software product development communities. This process will include definition of steps and events, roles and responsibilities, management checkpoints, and artifacts. Steps/events will include licensing as a transition vehicle. Artifacts will include:

- A software transition plan template.
- An implementation plan describing how the documented process will be put into operation at the SEI.
- A measurement plan describing how success of the documented process will be measured.
- A plan for testing the documented process outside the SEI.

4.1.1.7 Related TO&P Activities

There are currently no related TO&P activities in this area. In the future, TO&P may fund work to extend and apply the *Guide to Methods and Practices for Software Technology Transition Push* beyond the SEI. TO&P may also fund activities described in Section 4.1.2 not funded by core funds.

4.1.2 Proposed Add-On Activities

The following section describes proposed core add-on activities in the software technology transition area. The SEI is asking selected members of the software community to evaluate the relative attractiveness of these add-on proposals as possible elements of the final 1995 core program, as funding permits. Appendix A lists all baseline and add-on items.

The following activities build on accepted practice in leading edge organizations; they draw from and integrate for the software community a varied body of applied research; and they

translate both accepted practice and research into guidance for product developers and change agents within the software engineering community.

TT-1A Guide to Software Technology Transition Training and Continuing Education
The *Guide to Software Technology Transition Training and Continuing Education* will provide an overview of software technology transition, a curriculum design, instances of courses that satisfy the design, guidance on determining the best courses for individuals in specific software technology transition roles, and guidance on selecting suppliers of education or training materials. The curriculum design will consist of a design, three major tracks (each based on the life cycle of the major arena), and general course descriptions. The design builds on the Transition Models conceptual framework for software technology transition (CMU/SEI-93-TR-31) [Fowler 93a], as illustrated in Figure 4-1.

Objectives
With the increase in governmental pressure to reuse technology developed for defense, and more generally, to use technology to leverage U.S. competitiveness, focus on technology transition has increased, especially within the technology community that includes software. Many software engineers and managers now must transition technology, and do so quickly and systematically; yet they lack expertise, experience, and skills. Furthermore, the approaches they use are ad hoc, and the time required for software transition is far too lengthy. Understanding the processes within and between each of the three software transition arenas, the overlaps between them, and how best to manage these processes, provides a basis for translating concurrent engineering principles for software technology transition. In turn, this opens up the possibility of reducing the time it takes to move a software technology from an R&D setting into widespread use [Redwine 84] [Willis 83].

This reduction cannot occur, however, without skilled practitioners of technology transition. New conferences and continuing education opportunities in the area are proliferating, as everyone jumps on the bandwagon of this new "hot" topic. In other instances, useful courses are not labeled "software technology transition" and so are not selected. Those selecting training and education, or conferences and seminars, need criteria for choosing where to spend their time and money, and guidance on where to look.

Proposed Work
The *Guide to Software Technology Transition Training and Continuing Education* would guide the selection of software technology transition seminars, continuing professional education, training courses, and academic education (undergraduate and master's level). We will design a model curriculum for the various roles in software technology transition: software R&D managers and principal investigators, new software product developers, and change agents or individuals responsible for introducing software technologies within their organizations (see Figure 4-1). The curriculum will guide selections of courses for each role category.

Existing courses—both from the SEI and from elsewhere (e.g., the National Technology Transfer Center or its repository of technology transition training and education suppliers)—will be identified and categorized, with supplier contact information (we will note clearly that no endorsement is implied!). Where courses are not available, we will make rough estimates of cost to develop and identify possible sources for these. Finally, we will suggest criteria for selection of training and education vendors.

TT-2A Report on Best Practices in Software Technology Transition

Excellence in software transition practice means bringing software products to market faster and incorporating new technologies into building those products and doing everyday business. Leading organizations often have defined processes for software technology transition and pioneer the use of models and mechanisms. They set aggressive goals and benchmark their strategies against those of others. However, typically these companies do not share their software transition strategies in product development and technology implementation, because they consider these part of their competitive advantage.

Objectives

The SEI, as a neutral party, is in a unique position to assess and synthesize industry practice in software technology transition. The SEI can leverage this position to work with innovative organizations and tap their best practices in this area for the benefit of all. (Among others, Hewlett-Packard, Texas Instruments, AT&T Bell Labs, Schlumberger, and Motorola, as well as some tool vendors such as CaseWare have notable activity in the area of transition.) To date, SEI work in software technology transition has concentrated primarily on helping organizations adopt and implement new software technologies (see, for example, Section 4.3). The *Report On Best Practices in Software Technology Transition* builds on a strong understanding of organizational "pull" capability, and moves beyond that to grapple with software transition issues that face the software product development community.

Proposed Work

Developing the report will be a two-part activity. First, we will survey a representative sample of leading organizations to identify potential best software transition practice. Emphasis will be placed on expertise in new software product development and software technology adoption and implementation, with the latter keyed to levels of the CMM. After collecting the information through survey and interview, findings will be documented and integrated to produce a composite picture of best practice in software transition. [This is a refinement of the method used to produce the *Software Engineering Process Group (SEPG) Guide*, the basis for the creation of several thousand SEPGs.] The report will describe new software product development and software technology incorporation practice; we anticipate that further development and extension in future versions will include best practice in software R&D environments.

TT-3A Technology Transition Project Management Tool

Preliminary work has demonstrated the utility and efficiency of managing the introduction of individual new software technologies using project management techniques and tools. Change agents working to "pull" new software technologies into their organizations are able

to use this guidance to expedite the transition process. This process begins with early tasks such as vendor selection and pilot site selection and ranges to include later tasks such as setting up user support and rollout.

This task complements, extends, and enhances the work output described in Section 4.1.1.6, addressing the "pull" side of software technology transition for individual software technologies, particularly those at CMM Level 3 and above (for more on the introduction of software process improvement, see Section 4.3). It responds to requests from change agents within the software engineering community for more detailed direction in planning software technology transition.

Objectives

The objective of this activity is to create a software transition project management tool for change agents. This tool will guide planning for a transition effort; identify goals, constraints, and success criteria; define roles and relationships; guide trial use and rollout; and discuss training considerations. It is designed to feed data into common project management software, so that change agents can leverage the functionality of project management tools that they are already using.

Proposed Work

In developing this tool, we will build on and extend an early prototype created by the SEI. In the alpha test, this prototype proved to be instrumental in assisting the change agent with the introduction of software inspections in several projects in one large organization. Use of the prototype enabled inspections to be introduced and to be up and running smoothly in six weeks rather than in several months with uneven results.

We propose to:

- Provide the initial version of the tool to address any software technology along the four dimensions of transition shown in Figure 4-3: how individual behavior changes (including skill and knowledge); how the organization changes (for example, reward systems, policies, and practices); how technology affects people and organizations; and how the dimensions of people, organization, and technology affect the specific transition process.
- In the future, assess the viability and benefit of versions of the tool that address classes or categories of software technologies such as key practice areas (associated with the CMM).
- In the future, assess the viability and benefit of a version of the tool for product developers managing the process of new product development.

TT-4A Transition Assessment Instruments

Why are some software technologies and innovations adopted more rapidly than others? Notwithstanding economic value, the size of present and future markets, and technology standards, a set of product-related attributes appears to have an impact on the rate of technology adoption. These attributes include: (1) relative advantage, (2) compatibility, (3) complexity, (4) trialability, and (5) observability. Relative advantage, for example, is "the degree to which an innovation is perceived as being better than the idea that it supersedes." [Rogers 83] Trialabil-

ity is "the degree to which an innovation can be experimented with on a limited basis." Additional characteristics, including communicability and profitability, have also been identified [Tornatzky 90]. *Perceptions* of the attributes of innovations, not the attributes as classified by experts or change agents, affect their rate of adoption.

In addition, some organizations are more successful than others in designing software transition processes for technologies or products they develop or in introducing technologies into use.

Objectives

For most organizations, the software transition process ends at purchase, not use. The assessment instruments for software technology transition described here will codify existing knowledge and expertise on technology adoption, introduction, and incorporation, for the express purpose of helping developers, including software tool builders, to build products that are more readily transitioned into use. Moore [Moore 91] has offered guidance on producing the "whole product," a product that includes all adjunct services and related products (such as consulting, training, customer support, etc.). Leonard-Barton has suggested a process of "mutual adaptation" between organization and product, particularly in the early life of products [Leonard-Barton 88]. These sources plus our own work with *users* of software products indicate a need to educate vendors—especially in software start-up ventures—in issues of "transitionability." The result will be software products designed to be readily transitioned—a benefit to both those selling the products and those taking them in—and organizations more capable of transitioning products into use.

Proposed Work

We propose to develop a three-part assessment questionnaire for use by software engineers and their managers that will provide feedback on the readiness of a software technology for transition, the capability of an organization to produce transitionable technologies or products, and the capability of an organization to pull in, or introduce and incorporate, software products. As it is currently conceived, the assessment will be based on several related dimensions: (1) key product features, such as ease of use and divisibility (the extent to which incremental use is possible); (2) perception of attributes of the innovation (complexity, compatibility, etc.); and (3) organizational capability to "push" or "pull." The software community and its practitioners will benefit from an operationalization of this work focused on building and transitioning software products.

4.2 Education in Technology Transition

Education plays a significant role in technology transition. Our charter recognizes its importance as a mechanism for improving software engineering practice. The charter states:

> *The SEI shall evaluate, develop, and conduct courses and seminars that support transitioning evolving software engineering state of the art and practice. It shall also influence software engineering curricula development throughout the education community, industry, and government.*

We believe that:

- Properly educated people are better prepared to engineer software for increasingly complex systems.
- The education community has an important role in establishing software engineering as a recognized discipline and profession.

Three types of education are important to the development of a highly qualified software engineer:

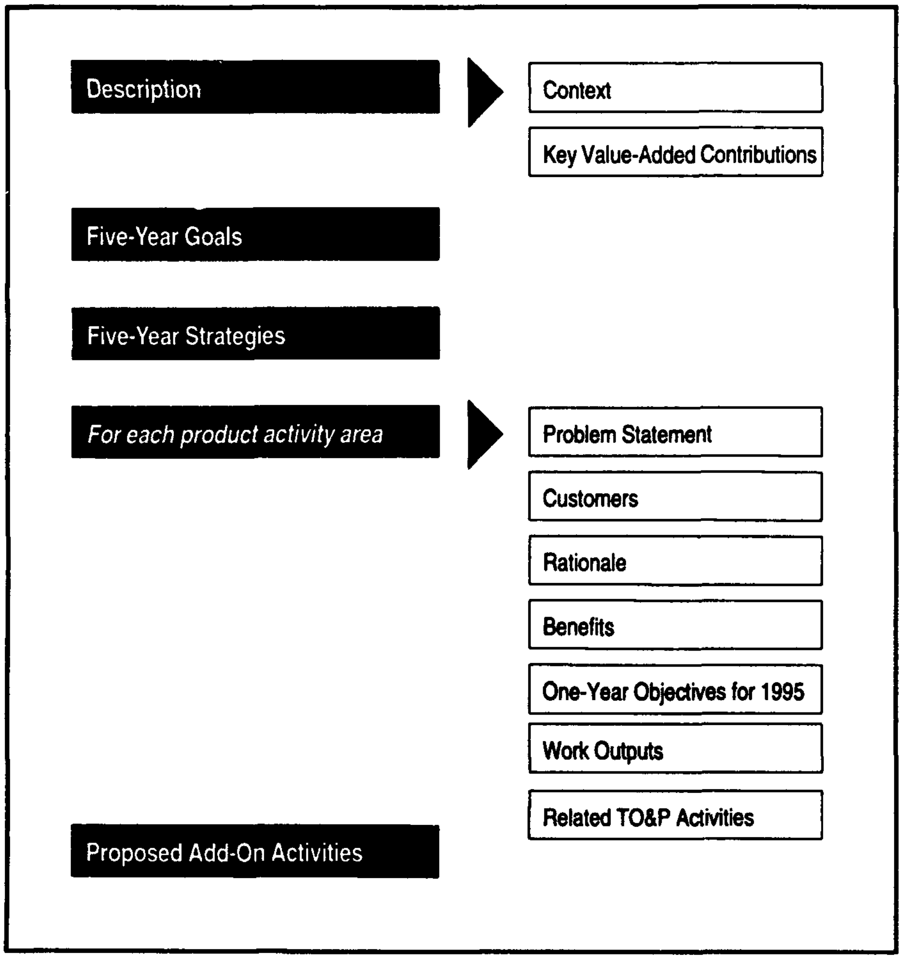
1. The initial education that prepares the engineer for entry into the profession; this is normally provided by the academic community in the form of bachelor's and master's degree programs.
2. Professional education that enables the engineer to stay current in a rapidly evolving discipline; this is provided through a variety of mechanisms, including in-house education programs, education vendors, university courses, and professional conferences and publications.
3. Education that enables an engineer or an organization to adopt and use specific new technologies; this education and associated training are normally provided in-house or by the technology vendor.

Currently, there are no undergraduate and only a limited number of graduate programs in software engineering in U.S. universities. Consequently, software engineers continue to enter the profession with inadequate education. This increases the burden on professional education, which must address deficiencies in preparation as well as advances in the discipline. Very large software organizations may have an internal professional education capability, but the majority of software engineers have few opportunities for professional education.

Our vision is a software engineering community in which all types of education are available and of high quality. In addition, we envision software engineering to be an accepted academic and professional discipline. To achieve our vision, SEI education activities seek to provide or influence all three types of software engineering education. In keeping with the charter, we focus our activities in two areas:

1. Developing and delivering educational products (see Section 4.2.4).
2. Helping to create an infrastructure that supports software engineering education (see Section 4.2.5)

The sections for this area of expertise are:



4.2.1 Description

4.2.1.1 Context

According to the recent report by the Blue Ribbon Panel [SEI 94], technology used in the DoD and the commercial sector remains far behind the state of the art. However, software engineering practice must be founded on a firm understanding of the technology. Understanding the technology is critical to installation, adoption, and eventual institutionalization. To improve the state of software engineering practice, it is imperative that software practitioners and managers have the requisite education and training.

Education is therefore a significant component of the SEI core competency in software technology transition. SEI educational products and services help to mature both the profession and the process of software engineering.

The SEI focuses on academic education as the principal means of preparing new software engineers for the profession; we provide products and services that educate practicing software engineers; and we offer products and services that motivate and enable strategic leaders to support software process improvement.

Educational Products Advisory Board. An Educational Products Advisory Board provides advice to the SEI on its activities in support of software engineering education. The board comprises two representatives each from government, industry, and academia, and meets twice a year.

This year, we will place less emphasis on academic curriculum development for master's programs than we have in the past. At the same time, we intend to increase our participation in the efforts of professional societies to define software engineering as a profession and to define academic curricula. We will also place more emphasis on delivery of our courses through the National Technological University (NTU), which broadcasts SEI courses by satellite, and less emphasis on train-the-trainer activities. To date, we have delivered five courses to a total of 151 students through NTU. As part of our effort to continuously improve the quality of our products and services, we will implement a development process for educational products that we defined in 1994 and ensure that all our products are developed using this defined process. This supports the Blue Ribbon Panel recommendation for more involvement by the education organization in the development of all SEI educational products.

SEI educational products serve an important function as transition mechanisms for process and technology work at the SEI. The CMM, RMA, domain-specific software architectures, organization capability development, and other SEI areas of focus are prominently portrayed within our educational products.

4.2.1.2 Key Value-Added Contributions

SEI strategies in education have increased the number of qualified software engineers. By providing curricula, courses, videos, and educational materials, we have improved foundational education in universities—education in such topics as software engineering process, software evolution, software generation, software maintenance, technical communication, software configuration management, and software quality assurance. By providing train-the-trainer courses and videos, we have improved in-house education. By delivering our courses through broad-based delivery mechanisms such as videotapes and satellite, we have provided practitioners in remote sites with education that they would not have been able to obtain otherwise. We deliver core and elective courses in the Carnegie Mellon University (CMU) Master of Software Engineering (MSE) program via NTU, enabling NTU to offer an MSE degree program with credits transferable to CMU for use in the one-year residence option of the CMU MSE degree. By offering courses that apply our own work in the CMM, risk, and metrics to the perspectives and concerns of decision makers, we have initiated significant investments in software process improvement.

The SEI also sponsors the SEI Conference on Software Engineering Education (CSEE), the only conference dedicated exclusively to software engineering education. Held in cooperation with the Association for Computing Machinery (ACM) and the IEEE Computer Society, this conference gives educators and others the opportunity to share experiences and expand their knowledge of software engineering education and training. The conferences include in-depth tutorials, formal presentations, exhibits, and many opportunities for informal discussion.

In the future, we plan to work with the ACM and IEEE to define the profession of software engineering and to help universities establish BS degree programs in software engineering.

The SEI, a trusted neutral source of advice and guidance, and the CMU School of Computer Science, a leader in computer science research and education, together constitute a unique environment for this work. Nowhere else is there such a favorable juxtaposition of resources for influencing software engineering education.

4.2.2 Five-Year Goals

Our goals are to assure that:

- The software engineering professional community has widespread access to high-quality software engineering education through traditional providers (academic institutions, in-house programs, and commercial vendors) making effective use of traditional and emerging delivery technologies.
- Educational products are based on the best available proven technologies and processes.

4.2.3 Five-Year Strategies

The strategy for accomplishing these goals is the following:

- Develop and deliver courses, courseware, and seminars to government, industry, and academic organizations. Maximize distribution through the use of satellite broadcasting where appropriate.
- Provide educational products and services to the academic community that significantly influence the continued growth of master's programs and the establishment of the first bachelor's programs in software engineering. Offer direct advice to universities on curriculum design and implementation.
- Develop and disseminate educational materials that facilitate the teaching of software engineering.
- Help university faculty and government/industry instructors develop their abilities to teach software engineering.
- Stimulate the growth of educational capabilities outside the academic community (i.e., in-house programs and commercial vendors).
- Incorporate leading edge technologies and processes into SEI educational products. Develop and deliver courses for strategic leaders to help leaders and decision makers meet their objectives with respect to transitioning these technologies into their organizations.

- Participate actively in, and provide technical leadership to, activities of the IEEE Computer Society and the ACM with respect to the establishment of a profession of software engineering and the development of software engineering model curricula.
- Develop and implement an internal SEI process for development of educational products that includes product selection based on customer needs and the use of measures to evaluate the effectiveness of educational products.

The Blue Ribbon Panel recommended that SEI personnel with expertise in education assume a greater advisory role in the development and deployment of software courses. A large percentage of the products currently transitioned by the SEI to government, industry, and academia are instructional products designed to provide education and training. Since so much SEI work is therefore dedicated to the development and delivery of instructional products, it is essential that the SEI have a defined and mature process for the development of instructional products.

ETRB Operation. Quality assurance is a significant part of this process. The Education and Training Review Board (ETRB) is the existing SEI quality assurance organization for instructional products. The ETRB not only ensures the quality of instructional products but, through its careful reviews, supports development by bringing to bear the varied expertise of the board members. The central reviewing focus in the ETRB provides a greatly needed check on both the consistency and sound instructional merit of SEI instructional products.

Educational Process. In 1994, a quality improvement process (QIP) was initiated to define an internal process for developing educational products. The process defined by this QIP will improve product selection, customer needs analysis, collection and use of customer feedback, and the production of courses, educational materials, videotapes, multimedia education packages, and model curricula. The process will be linked to the ETRB and will include instructional design support. It will be implemented in 1995.

Education 10th Year Retrospective. In connection with this strategy to implement an internal process that focuses on customer needs, we will hold a 10th-year retrospective on software engineering education. This will be an invitation-only workshop to be held on the 10th anniversary of the kickoff meeting of the SEI Education Program in May 1995. The purpose of the workshop will be to examine and evaluate the directions and activities set 10 years earlier and to set some directions and activities for the next decade. The attendees will comprise a subset of those who attended the original May 1985 workshop and the February 1986 SEI-sponsored conference, along with all the current members of the Education Advisory Board.

4.2.4 Educational Product Development and Delivery

This section discusses the product activity area related to educational product development and delivery.

4.2.4.1 Problem Statement

The field of software engineering suffers from a shortage of instructors with appropriate understanding of the technology and process of software development. As a result, the defense community and other software-intensive endeavors continue to rely on inadequately educated personnel to develop software. This shortage of qualified educators creates a critical need for high-quality, cost-effective course materials to aid instructors.

Because of constraints of organizational policy or geography and the scarcity of university programs, attending a university program in software engineering is a rare privilege. In-house opportunities usually focus on tool or methodology training; however, tool and methodology training are less effective without education in the necessary foundational concepts of software engineering. Initiatives to introduce a new technology into practice have repeatedly failed because new practitioners lack context and the underlying concepts for the new technology, whether it be process- or product-related. In-house development of courses in these broader areas is expensive for government and industry, and hard to justify. Thus software-intensive organizations also need quality course materials.

Strategic leaders can benefit from SEI educational products that support their efforts at instigating software process improvement within their organizations. Too often, leaders are inhibited in these efforts. A survey of CMM-based education and training cited the following key inhibitors of process improvement in organizations:

- Lack of management commitment.
- Inability to provide return on investment data to justify investment in process improvement initiatives.
- Lack of knowledge about how to manage software process improvement.

4.2.4.2 Customers

Customers for academic and practitioner courseware are:

- Educators implementing software engineering programs.
- Software engineering practitioners, including those who have returned to the academic community for further education.
- Government/industry educators.

Customers for leadership education are:

- Leaders and key decision makers in the software community.
- Government and industry executives with strategic responsibility for software.
- Members of SEPGs.
- Managers of software process improvement efforts.
- Process improvement champions.
- Others whose decisions have strategic impact on their organizations.

4.2.4.3 Rationale

As shown in Figure 4-5, academic education is a traditional prerequisite to professional standing. SEI initiatives in academic education enhance and accelerate foundational software engineering education in the academic community by serving as a central source for the creation and dissemination of high-quality educational products.

However, many software practitioners do not have formal background in software engineering or even computer science, and university programs, although they are growing, are not available to all practitioners. Practitioners are more likely to take advantage of educational opportunities if they are available locally, either delivered in-house by trained educators or delivered by satellite. The SEI produces videos, course materials, and satellite courses to bring educational opportunities to software practitioners.

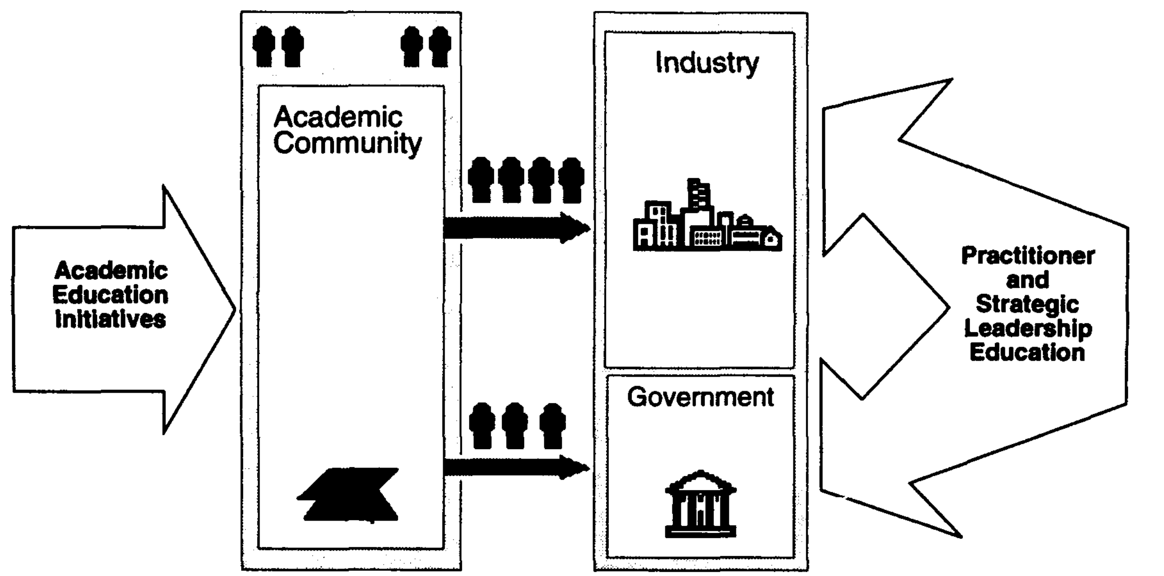


Figure 4-5: Program Rationale

To improve the state of the practice of software engineering, we also need to work with organizations in bridging the gap between tactical initiatives for software process improvement and strategic corporate goals. Software technology transition requires support from the leaders and decision makers of the organization, who need guidance in developing and implementing strategic plans for software process improvement.

Champions and sponsors of software process improvement need education on proven methods and experience. Technical arguments are not effective in obtaining funding and managerial action unless the arguments for change are couched in terms of executive concerns: benefits to the organization's profit margins, survival in a competitive environment, increased

customer satisfaction, return on investment, lower costs/higher productivity, ability to meet commitments, etc.

Through leadership education, the SEI builds management support for software process improvement. Since about 1992, students coming to our executive courses (which we now refer to as "leadership" courses) have reflected the maturing of the software community. They typically have already participated in a software process assessment and have formed a SEPG. Most have a general understanding of the CMM. Although their organizations are ready to respond to assessment findings, the organizational leaders need guidance in how to justify and manage a software process improvement plan. They also need a more strategic view of what actions they should take over a five-year period (not just implementing action plans based on assessment findings) and how this long-term strategy complements their business goals. Leadership education meets this need.

4.2.4.4 Benefits

Figure 4-6 depicts the education trends over the next four years.

Key Items	State of practice as of:			Impact/Metrics
	Today	+2 Years	+4 Years	
Courses, courseware, and seminars for government, industry, and academic communities.	Much SEI courseware and many SEI seminars available, many through satellite access.	Significant satellite access; some computer-based multimedia products available.	Satellite access over several networks; desktop availability common.	Widespread access to high-quality courses, courseware, and seminars.
Model curricula that influence the continued growth of master's programs and establishment of the first bachelor's programs in software engineering.	SEI graduate curriculum published in 1989.	SEI undergraduate curriculum exists.	ACM/IEEE undergraduate curriculum exists, with SEI influence.	Widespread use of model curricula; high-quality software engineering education widely offered at bachelor's and master's levels.
Educational materials that facilitate the teaching of software engineering.	Much courseware exists (videotaped lectures and instructor materials).	More courseware marketed to educators.	Significant emphasis on satellite and desktop delivery	Widespread access to educational materials.
Services that help instructors develop their abilities to teach software engineering.	Limited train-the-trainer sessions; the SEI CSEE is the premier conference for software engineering educators.	SEI provides more advice and guidance on curricula development and training plans/implementation.	Government/industry organizations have mature view of their educational needs and can match needs to resources.	Widespread access to well-educated, effective instructors.

Figure 4-6: Trends in Education in Technology Transition

Key Items	State of practice as of:			Impact/Metrics
	Today	+2 Years	+4 Years	
Educational capabilities outside the academic community (i.e., in-house programs and commercial vendors).	Clients host SEI courses at home locations; clients use SEI courseware; clients receive SEI courses/seminars on NTU.	Significant SEI course delivery on NTU; SEI advice and guidance on curricula development and training plans/implementation.	SEI courses on several satellite networks; desktop availability common; government/industry organizations have mature view of needs and can match needs to resources.	Widespread access to high-quality in-house programs and programs offered by commercial vendors.
Educational products that incorporate leading edge technologies and processes.	Customers have access to current models, experience data, and proven technologies.	More customers have access.	Most customers have access.	Educational products are based on the best available proven technologies and processes.
Courses for strategic leaders.	5-course curriculum developed and delivered; 2 courses available on NTU; on-site delivery available.	All 5 courses available on NTU; on-site delivery available.	All 5 courses available on several satellite networks.	Widespread access to courses for strategic leaders that help leaders and decision makers meet their objectives.
Internal SEI process for development of educational products.	Internal process defined.	Internal process applied across the SEI, shared with clients, and improved with experience.	Internal process applied across the SEI, shared, and improved.	Process includes product selection based on customer needs and the use of measures to evaluate product effectiveness.

Figure 4-6: Trends in Education in Technology Transition (Continued)

By leveraging our efforts through distribution agents (educators, satellite, computers), the SEI is able to reach large numbers of students with a small SEI staff. By featuring subject matter experts in our courses who would not otherwise be available to most organizations, SEI courses enable practitioners to receive high-quality software engineering education. Prestigious lecturers in SEI courses include:

- Robert Charette (author of *Software Engineering Risk Analysis*)
- Peter Coad (co-author of *Object-Oriented Analysis*)
- Alan Davis (author of *Software Requirements: Analysis and Specification*)
- Norm Gibbs (recent recipient of the ACM SIGCSE Award for Outstanding Contributions in Computer Science Education)
- Hassan Gomaa (author of *Software Design Methods for Concurrent and Real-Time Systems*)
- John Musa (co-author of *Software Reliability*)
- Herb Simon (Nobel Laureate, the Richard King Mellon University Professor of Computer Science and Psychology at CMU).

Executive education has led to increased management sponsorship and support of software process improvement efforts that improve the state of the practice. Our leadership courses encourage strategic leaders and decision makers to:

- Baseline current practice.
- Set improvement goals.
- Create a balanced portfolio of improvement efforts offering short-term and long-term benefits.
- Manage risks.
- Measure benefits of improvement.

The original intent of the executive curriculum was to enhance awareness among key decision makers of the benefits of embarking on software process improvement activities. Sixty percent of surveyed attendees attribute positive influence on their process improvement efforts to attendance of an SEI executive course. "Software: Profit Through Process Improvement" in particular has been credited with motivating companies to instigate software process assessments and establish SEPGs. Through SEI executive courses, some executives have for the first time become aware of the scope of SEI activities and have become regular participants in SEI events. Additionally companies have become more committed to participating in SEI education/training classes.

4.2.4.5 One-Year Objectives for 1995

Participate in the CMU MSE Program. SEI participation in the MSE program enables us to gain a better understanding of curriculum issues and establishes a timely process for maintaining the currency of the SEI academic courses. Delivery of CMU courses in the SEI studio simultaneously produces videotaped lectures for academic use and satellite broadcast via NTU to a remote audience.

Use NTU as a delivery channel for academic, practitioner, and leadership courses. Satellite distribution affords us a wider audience and access to the NTU marketing network. Through telephone, electronic mail, and FAX facilities, we maintain interaction with students and provide the feedback that is so essential to learning.

Increase penetration in the community of SEI courseware, videotapes, and educational materials. The internal process for developing educational products will be implemented, and this process will add to our success in building high-quality products by emphasizing product selection, analysis of customer needs, and collection and use of customer feedback.

Maintain the currency of leadership courses and deliver them in open enrollment classes at SEI locations. We will continue to deliver the five courses in the executive curriculum, which we now call the "leadership" curriculum, at SEI locations as indicated by demand. These courses are:

- Software: Profit Through Process Improvement
- Software Quality Improvement

- Software Productivity Improvement
- Software Risk Management
- Managing Software Development with Metrics.

All deliveries are funded through cost recovery.

Deliver leadership courses and tailored versions of practitioner courses at client sites, as negotiated. On-site delivery is economical for the client and permits broader participation in course offerings. It enables the instructor to focus on the needs of a specific audience.

The SEI will offer tailored mini-courses derived from the practitioner courses as negotiated for delivery at the client's site. These courses are:

- Software Project Management
- Software Requirements Engineering
- Software Design

To date, on-site offerings have been delivered only to government organizations, but there is interest among the software process improvement networks (SPINs).

4.2.4.6 Work Outputs

All the work described in this section is ongoing work begun in 1994 or previous years and extended into 1995.

MSE core and elective courses. We will update one-third of the courses each year and deliver them through NTU. These courses form the foundation of an MSE degree program that is offered by NTU, and some of them can be transferred to CMU to use in the one-year residence option of the CMU MSE degree. They are based on the continuing curriculum work done at the SEI. Teaching the courses on NTU enables us to directly reach more of our targeted customers. NTU receive sites are at most U.S. Government R&D centers such as the Naval Air Center, Naval Surface Weapons Center, Army Automotive and Tank Command, and nearly all corporations heavily involved in software development. Also, teaching the ccurses on NTU results in recovery of production costs.

Professional Education and Leadership Series: *Software: Profit Through Process Improvement, Software Quality Improvement, and Software Design.* These courses will continue to be delivered through NTU. We will maintain the currency of the courses so that their content reflects the latest SEI work and the latest government/industry experience. During 1994 the SEI broadcast two offerings of Software: Profit Through Process Improvement, two offerings of Software Quality Improvement, and one mini-course derived from the practitioner course Software Design. In addition to recovering some of the broadcast cost through a percentage of the NTU tuition, videotapes of the broadcast are produced and marketed by NTU for additional cost recovery by the SEI.

The 8th Conference on Software Engineering Education (March-April 1995). This is the only conference devoted specifically to software engineering education. Educators, trainers, managers, and administrators gather together to exchange ideas about how to enhance software engineering training and education. The CSEE attracts attendees from government, industry, and academia. Its purpose is to influence educational directions, stimulate new approaches, promote collaboration, and generate interactive exchanges among all educational stakeholders. The 1995 conference will focus on the impact of education on practice, education and training goals, people, process, technology, industry-academia collaboration, training and education management, and delivery. The conference is sponsored by the SEI in cooperation with the IEEE Computer Society; ACM cooperation is pending. The proceedings will be published by Springer-Verlag as part of the *Lecture Notes in Computer Science* series.

Maintenance of the existing portfolio of courseware packages and videotapes of the Professional Education practitioner courses, Academic Series courses, and the Technology Series.

The Professional Education practitioner products are derived from the following courses:

- Software Project Management
- Software Requirements Engineering
- Software Design

These products comprise 73 videotapes and supporting courseware.

The Academic Series products are video-based courses that include:

- Constructing Reusable Software with Ada
- Formal Methods in Software Engineering
- Managing Software Development
- Software Design
- Software Design, Creation, and Maintenance
- Software Product Development
- Software Specification
- Software Verification and Validation

The Technology Series consists of stand-alone videotapes:

- Executive Leadership for Software
- Applying Software Engineering Skills to Writing
- Software and Some Lessons from Engineering
- Risk Management Culture
- Motivation for Software Risk Management
- Up the Down Escalator: A Dynamic Three-Dimensional Model for Managing Risk
- An Introduction to Rate Monotonic Analysis (RMA)

These materials provide a wealth of information to educators, students, and practitioners.

4.2.4.7 Related TO&P Activities

There are no related TO&P activities in this area.

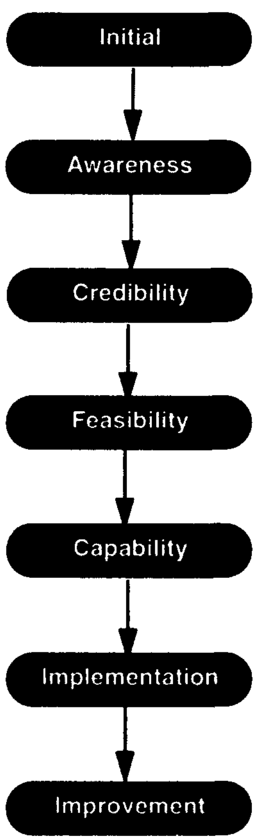
4.2.5 Educational Infrastructure

This section discusses the product activity area related to educational infrastructure.

4.2.5.1 Problem Statement

Currently there are no undergraduate and only a few graduate programs in software engineering in U.S. universities, so software engineers continue to enter the profession with inadequate education. The SEI has been working since 1985 to accelerate the growth of an infrastructure for software engineering education within the United States.

Infrastructure work addresses broad organizational change. Figure 4-7 represents a model of the development of academic programs in universities. The bubbles in the graphic represent the phases through which a school passes in creating a new program.



No thought has been given to software engineering or to software engineering education.

There is an awareness that software engineering is potentially something different from programming, computer science, and computer engineering.

There is a belief that software engineering is different from programming and computer science, and that it might be taught as a separate program.

There is a belief that a separate software engineering degree program is possible within the university.

There are sufficient university resources available to provide a software engineering degree program in the university.

A software engineering degree program is being put in place.

The software engineering degree program is being improved, both in content and in teaching effectiveness.

Figure 4-7: Phases in the Development of Academic Programs

The model has two important uses:

1. It suggests the kinds of barriers that inhibit an organization moving from one phase in the model to the next; this in turn allows us to identify the kinds of SEI interventions that can help remove those barriers.
2. It gives us a framework for assessing a customer to determine what products and services to provide to that customer.

4.2.5.2 Customers

Customers for SEI work in building an educational infrastructure include:

- Faculty and students in colleges and universities
- Professional societies
- Accreditation agencies
- Textbook authors and publishers
- Employers of software engineers

4.2.5.3 Rationale

The need for highly qualified software engineers is pervasive—it touches all segments of the software engineering community. The government and industrial software communities will continue to need competent software engineers. It is substantially more cost-effective for software organizations to hire people who have been educated as software engineers and can be productive almost immediately than it is to hire people who will need years of additional education (at company expense). Thus developing the capabilities of the U.S. academic community to produce high quality software engineers is an important part of the SEI strategy to improve software engineering practice.

The academic community changes slowly. Despite years of publicity about the "software crisis," there are still no undergraduate software engineering programs in U.S. universities. The SEI is uniquely positioned to catalyze and accelerate the growth of academic software engineering education. By doing so, the SEI may be able to help the academic community reach the point of producing an adequate number of software engineers in 10 to 15 years less than would be the case without SEI help.

To change the academic infrastructure requires that we understand the components of that infrastructure and their modes of influence and interaction, represented graphically in Figure 4-8. Each of the boxes and arrows represents an opportunity for exerting SEI influence to improve software engineering education. SEI work in building an educational infrastructure addresses many of these (although not all in 1995).

Work on the educational infrastructure is essential to maintaining the SEI core competency in education. This competency allows the SEI to respond quickly and meaningfully to a wide range of unanticipated customer requests for information, advice, or collaboration.

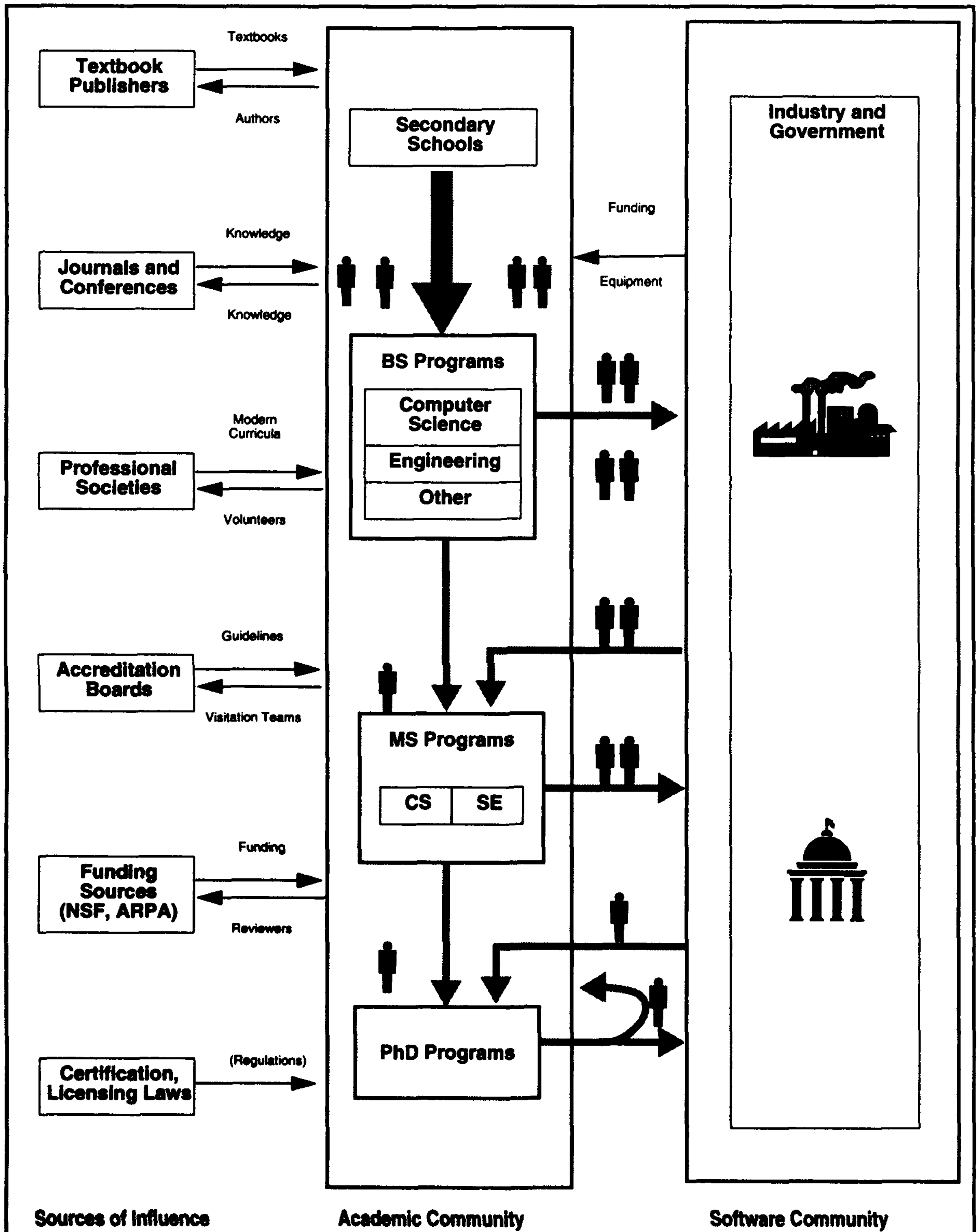


Figure 4-8: Interactions Within the Academic Infrastructure

4.2.5.4 Benefits

Figure 4-6 depicts the education trends over the next four years.

SEI work on the educational infrastructure is designed to enable the academic community to provide continuing education to several thousand working professionals each year and to produce about 20,000 new software engineers a year by the year 2015. Work performed so far has been well received and influential. As shown in Figure 4-9, two-thirds of the schools currently offering software engineering programs acknowledge SEI technical reports, curriculum modules, and similar publications as significant influences on their decisions to start a program and on the design of their curricula. Even among schools that do not currently acknowledge SEI influence on their curricula, having based their designs on other models (such as the curriculum at the Wang Institute of Graduate Studies), SEI efforts are appreciated and well received.

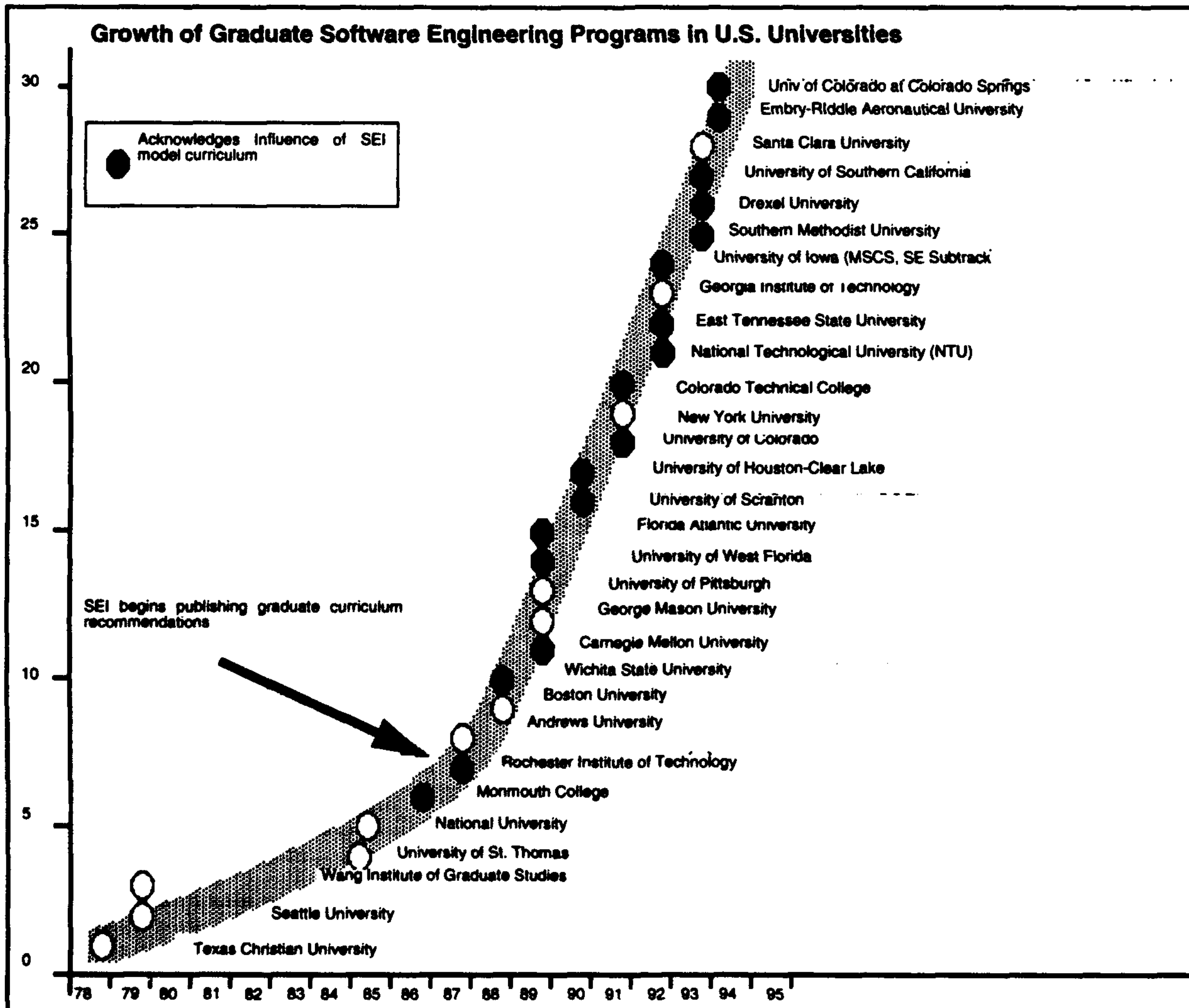


Figure 4-9: Influence of SEI Curriculum Reports

This work indirectly provides for the transition of many technologies into the software community. The curriculum recommendations and educational materials produced by this work incorporate, wherever possible, the most recent work of SEI technology projects. This further facilitates the transition of the results of those projects. We also plan to cooperate with education-related work of the IEEE Computer Society and the ACM. Ongoing actions of the professional societies to promote the software engineering profession will increase the receptiveness of the academic community to the products and services proposed. Infrastructure work also contributes indirectly to the knowledge base, in that the growth of academic programs in software engineering will produce a corresponding growth in the number of faculty conducting basic and applied research in software engineering.

The SEI is uniquely qualified to perform this work. We have already established an international reputation for leadership in academic software engineering education. No other organization is doing this kind of work. Many of the barriers to the creation of new academic programs are the same at each school. The SEI can provide both generic and tailored solutions so that individual faculty members at each university do not have to work in isolation to overcome these barriers. Thus the work of a few SEI staff members can more quickly enable software engineering to be taught at hundreds of universities, resulting in tens of thousands of better qualified professionals.

4.2.5.5 One-Year Objectives for 1995

Our objectives are to help educators overcome the barriers that inhibit their organizations from developing high-quality academic programs and thereby increase the quality and quantity of software engineering in the academic community.

The major components of the work are:

- Research to identify and quantify the needs of the software community that can be satisfied by the academic community.
- Translation of those needs into influential products and services for the academic community (meaning products and services that help remove the barriers that inhibit an organization moving from one phase to the next in the model shown in Figure 4-7).
- Widespread delivery of those products and services to appropriate people and institutions.
- Monitoring the development of the academic infrastructure to plan future work, validating the effectiveness of past work, and informing the software community of progress.

Activities in support of these objectives include:

Conduct faculty development workshops in conjunction with conferences. In recent years, such workshops have been held at the SEI Conference on Software Engineering Education and at the ACM Special Interest Group on Computer Science Education (SIGCSE) Technical Symposium on Computer Science Education. At least one full day of workshops is planned for 1995. The content of the workshops will be derived from recently completed technical reports and educational materials and will help educators improve their capability for delivering high-quality software engineering education.

Visit universities to promote software engineering education and to consult on curriculum design. Recently, 10 to 15 such visits have been made each year. The SEI has also hosted meetings of university faculty to collaborate on curriculum design. In a few cases, SEI staff sit on university, college, or departmental industry advisory committees for software engineering programs. This kind of direct contact with customers helps overcome barriers to credibility, feasibility, implementation, and improvement, and has proved extremely valuable for all parties—the customers gain the benefit of SEI knowledge and experience, and we gain a better understanding of the needs of the customer. This work will continue in 1995.

Host visiting scientists and university faculty on sabbatical leave. We have already hosted more than 60 visitors in SEI education-oriented projects. Immersing faculty in the SEI technology culture has proved to be an effective transition mechanism. Visiting scientists can immediately incorporate the knowledge gained into their courses and materials, which helps to overcome feasibility, capability, and improvement barriers in their organizations.

Collaborate with IEEE Computer Society and ACM to establish the software engineering profession. The two professional societies have recently begun efforts that will significantly influence the growth of the software engineering profession. These efforts will include development of model academic curricula in software engineering, which raise the credibility and feasibility of software engineering academic programs. SEI participation in these efforts will also increase the visibility and credibility of SEI curriculum work.

Influence the influential. We will continue our efforts to overcome barriers to the development of academic programs by convincing influential people and organizations of the importance of software engineering education. This includes activities such as:

- Meeting with editors from textbook publishing companies to explain the importance of software engineering and to help them identify topics and authors for future textbooks.
- Presenting papers and participating in panel discussions at conferences to promote awareness and understanding of software engineering as an academic discipline.
- Publishing papers on software engineering curriculum design in major journals.
- Participating in curriculum efforts of the professional societies.
- Meeting with the chairs of the education-related boards and committees of the professional societies.
- Advising Advanced Research Projects Agency (ARPA), Defense Information Systems Agency (DISA), and the National Science Foundation (NSF) on issues related to funding for software engineering education.
- Providing information to university administrators (deans and above) to convince them to support software engineering programs at their schools.
- Meeting with members of accreditation boards to promote appropriate accreditation procedures for emerging software engineering programs.
- Corresponding with state legislators on issues related to licensing of software engineers.

4.2.5.6 Work Outputs

Two technical reports on important issues in software engineering education (yearly). Topics of technical reports include curriculum design, course content, pedagogy, laboratories and project work, faculty development, professional ethics, certification and licensing, academic program accreditation, funding opportunities, research opportunities, etc.

Proposed topics for 1995 are:

- Goals and Objectives of Undergraduate Software Engineering Education
- Laboratories in Software Engineering Education

Six educational materials (EM) packages (yearly). Educational materials packages include such materials as lecture notes, course descriptions, examples of software work products, bibliographies, examples, case studies, exercises, examinations, laboratory materials, transparency masters, etc. The topics of these EMs are selected two or three times a year, depending on several factors: recent results from SEI technology projects, materials submitted by software engineering educators in universities, and the identification of important topics that are not currently covered in textbooks.

Proposed topics for 1995 are:

- Experimental Methods for Software Engineers
- Lecture Notes on the Cleanroom Development Method
- Examples of Professional Codes of Ethics
- Maintenance Exercises: An Ada Software Size Tool
- Examples of Software Architectures
- Examples of System Modeling

4.2.5.7 Related TO&P Activities

There are no related TO&P activities in this area.

4.2.6 Proposed Add-On Activities

The following section describes proposed core add-on activities in education. The SEI is asking selected members of the software community to evaluate the relative attractiveness of these add-on proposals as possible elements of the final 1995 core program, as funding permits. The proposals are grouped according to the product activity areas within this area of expertise. Appendix A lists all baseline and add-on items.

4.2.6.1 Educational Product Development and Delivery

E-1A Software Systems Engineering Course

There is considerable feedback from the community that a course on this topic is needed. One industry group has proposed an entire curriculum on this topic. The SEI would fill a void in connecting systems engineering discipline with software engineering. Currently there are no products or services from the SEI that address this critical issue. The course will examine the

considerations necessary for the total engineering of a product, discuss how to make hardware/software tradeoffs, and discuss how to define interfaces between hardware and software. The course would be designed to meet the needs of software engineering practitioners with systems engineering responsibilities.

Objectives

By the conclusion of the course the student will be able to:

- Use at least two methods of system requirements elicitation.
- Perform hardware/software/process requirements allocation.
- Define software interfaces to hardware and users.
- Develop a system verification and validation plan.
- Sequence hardware/software component development and delivery.

Proposed Work

A proposed outline for this course is as follows:

- The roles of software and systems engineering.
- Methods of system requirements elicitation, including analysis, modeling, documenting, and verifying.
- Allocation of requirements to components.
- Reuse of software components.
- Designing interfaces to heterogenous components.
- Quality issues, acceptance testing, and other forms of validation.
- Scheduling development.
- Post-deployment support issues.

E-2A Real-Time Design Course

This course is a logical elective to offer in the SEI software engineering video series: it directly addresses the needs of the mission-critical systems community. The course will examine several approaches to real-time design and feature SEI products such as work in domain-specific software architectures and RMA.

Objectives

By the conclusion of the course the student will be able to:

- Define real-time software requirements in terms of platform constraints.
- Specify and design software components for handling discrete external events and discrete and continuous external data values.
- Design typical data filters such as first-order filters, threshold filters, lag filters, can filters, etc., to solve signal processing requirements.
- Describe the components of a real-time virtual architecture consisting of processes, communication, synchronization and timing mechanisms, and virtual devices.
- Prepare design structures using the virtual real-time architecture notions to describe real-time system designs.
- Describe the basic design principles for "gracefully degradable" (soft) deadline-driven real-time software.

- Prepare software designs with fault detection and fault-tolerant mechanisms.
- Prepare software designs implementing cyclic executive and periodic scheduling approaches.

Proposed Work

This course will not endorse any design method in particular, nor will it provide a survey of existing methods, techniques, and tools. Rather, the course will concentrate on design principles and techniques found to produce better results, and hence may draw from several existing methodologies.

A proposed outline for this course is as follows:

- Real-time systems, application areas, operational environments.
- Real-time requirements: functional requirements, reliability requirements, timing constraints.
- External interfaces: data acquisition, data (signal) processing.
- Timing and fault-tolerant control: time management, fault management.
- Architectures and design approaches: system architectures, software structures.
- Real-time concurrent programming: implementing a real-time virtual architecture; modern concurrent programming languages.
- Distribution and other advanced issues (concurrent object-oriented systems).

E-3A Open Systems Course

Because of current economic and policy forces (e.g., defense downsizing, new Secretary of Defense policies, and proposed changes in procurement regulations), acquisition agencies need to become more knowledgable about open systems concepts. This change affects all levels in the acquisition chain, from the highest offices in the DoD through the contractors and subcontractors that support them. There is widespread interest in a course on open systems. While the commercial marketplace offers such courses, these are often limited or biased in their vision of open systems, and they do not address the needs of program management offices.

A prototype open systems course was produced for the U.S. Navy Space and Naval Warfare Systems Command (SPAWAR) under technical objectives and plans (TO&P) 2-164. By agreement with the customer, this was a *prototype* to be presented only twice, with updates made to the course materials based on participant comments. Although the SPAWAR sponsor believes that this course will have a tremendous impact on the community and should be made available to as wide an audience as possible, funding was not provided for the additional effort required to make the course suitable for broader dissemination. The feedback from the dry run of the course was very positive. While there has been no attempt to advertise the availability of the course, interest has nevertheless been expressed by such organizations as the Office of the Secretary of Defense, the Defense Information Systems Agency, the Industrial College of the Armed Forces, the National Institute of Standards and Technology, the Naval Postgraduate School, and some industry groups, including Texas Instruments and a publisher of open systems information.

Objectives

The objectives of the course are to help the students:

- Understand basic terms and concepts.
- Understand program management issues related to the use of open systems.
- Understand both the potential benefits of open systems and the difficulties in creating systems based on open system components.
- Feel better equipped to deal with the paradigm shift associated with the use of open systems.
- Recognize that there are no easy solutions, and that open systems are not a silver bullet.

Proposed Work

- Complete the evolution of the course from a prototype to a finished product with the help of the ETRB. Use the prototype course materials and experience to produce improved course presentation materials, better exercises, an instructor's guide, and "train-the-trainer" planning.
- Tailor the course for various audiences and purposes. The course material could be linked to other SEI efforts, such as appraisal methods and the Software Engineering Improvement Framework. The work also has strong relationships with domain-specific architectures, reengineering, and reuse.
- Develop additional related courses and/or course modules as demand and resources allow.

The course covers the following topics:

- The paradigm shift caused by open systems
- Basics
- Standards
- Non-development items and commercial off-the-shelf (COTS) products
- Elements of an open systems approach
- Current practice
- Impacts of changes
- Program management office considerations
- Managing the transition
- System considerations
- Evidence
- Study exercise in time synchronization

E-4A Expansion of Leadership Series to NTU

In 1994, the SEI broadcast through NTU two executive courses and one mini-course derived from a practitioner course:

- Software: Profit Through Process Improvement (twice)
- Software Quality Improvement (twice)
- Software Design mini-course (once)

The continuation of these NTU offerings are included in the 1995 objectives.

Objectives

NTU satellite broadcasts help us distribute our products to a broad audience. They provide us with an established marketing network and enable us to deliver our courses more economically to our clients. Videotapes captured during the broadcasts are marketed by NTU. In addition to providing us with tuition to recover some of the costs, this gives us another source of SEI visibility and cost recovery.

Proposed Work

We propose to expand NTU short course offerings to include the three remaining leadership courses and an additional practitioner mini-course:

- Software Productivity Improvement (twice)
- Software Risk Management (twice)
- Managing Software Development with Metrics (twice)
- An additional mini-course derived from either Software Design or Software Requirements Engineering (twice)

Producing leadership courses for NTU entails maintaining the currency of all materials, restructuring the course to fit NTU broadcasting schedules and formats, creating NTU marketing collateral information, coordinating with NTU, creating promotional videos, paying studio and satellite fees, and providing instructor effort and general support.

Delivery of mini-courses derived from existing practitioner courses requires design of the mini-course using existing materials, creation of NTU marketing collateral information, coordination with NTU, creation of promotional videos, studio and satellite fees, and instructor effort and general support.

E-5A CD-ROM Packaging of Practitioner Series

Because of corporate and defense downsizing and the need for more flexible workers, organizations with limited budgets require more education/training for employees. For many, traditional classroom instruction is too expensive and seldom timely. Clients want to be able to receive training upon demand.

To respond to these needs, the SEI is preparing to become a supplier of digitized, multimedia courseware in software engineering subjects. In 1994 we purchased equipment and authoring tools and are in the process of producing a prototype of a self-paced computer-based course packaged on CD-ROM. Computer-based training on CD-ROM is relatively inexpensive, serves multiple students, and can be used at the student's convenience. Most major organizations now have equipment for multimedia learning stations, and the equipment is becoming less expensive. Major conferences are now distributing their proceedings on CD-ROM. CDs are inexpensive to reproduce and far more compact than our current courseware packaging. Once digitized for CD-ROM, educational materials could be transmitted over networks.

Objectives

- Leverage past efforts and create new products for the changing technological climate.
- Promote active learning by incorporating advanced instructional design technology and cognitive theory into courseware.
- Provide a vehicle for just-in-time learning.
- Increase the availability of SEI courses.

Proposed Work

In 1995 and beyond, we propose to build on our experience base and create additional CD-ROM products, using existing course videos and materials. We propose to:

- Package course materials, video, audio, and reference material for Software Requirements Engineering, Software Design, Risk Management, and Introduction to the Capability Maturity Model (CMM) into integrated hypermedia systems.
- Prototype linkages between CD-ROM development and network course delivery.

E-6A Support for Level 3 Key Process Area (KPA) on Training

The CMM has established an industry-wide interest in software process improvement. More and more organizations are reporting achievement of level 2 appraisals and are focusing on the KPAs defined for level 3. The training KPA at level 3 addresses the establishment of a mature and managed training program for the organization. Organizations naturally look to the SEI for guidance in maturing their training capability.

Objectives

Our objective in this work is to serve as a neutral source of advice and guidance in helping organizations to:

- Determine their training needs.
- Plan and implement curricula.
- Establish the infrastructure to deliver and administer courses.
- Evaluate commercially available education/training courses against their needs.

Proposed Work

The following work is proposed to provide products and services to client who are seeking to improve their training capability:

- Provide advice and guidance to government and industry organizations on curriculum issues and training infrastructure to help them satisfy the CMM level 3 training KPA. We have been providing advice on training needs since 1988, but without an overall strategy or product suite to present to organizations. Potential clients for this service include Texas Instruments, Lockheed, Motorola, Defense Logistics Agency, the Air Force Civilian S&T Committee, and the Internal Revenue Service.
- Keep the *Directory of Industry and University Collaborations with a Focus on Software Engineering Education* current. This document was created in 1994. Many of the collaborations mentioned in the directory are outgrowths of SPINs. We have visited several of these groups, and we must maintain and expand our relationships with them. Some of these groups perform needs analysis and design curricula.

- Expand the distribution of Professional Education courses to one or more new partners (satellite broadcasters, professional consortia, vendors). A strong potential is IEEE Videoconferencing through University Television at California State at Long Beach (CSLB). CSLB is also the distribution site for courses used by the Software Engineering Forum for Training, an outgrowth of the Southern California SPIN.
- Develop an instrument to help clients select courses from vendor offerings. The CMM training program KPA includes procuring training to address identified needs. The instrument could be a document or a hypertext artifact.

4.2.6.2 Educational Infrastructure

E-7A Leadership on IEEE/ACM Task Force On Software Engineering Profession

The two largest associations of computer professionals in the world, the IEEE Computer Society (CS) and the ACM, have established a Joint Steering Committee to evaluate, plan, and coordinate actions necessary to establish software engineering as a profession. The SEI plays a leading role in this activity, with one member of the SEI chairing and two members serving in the eight-person steering committee. The steering committee has chartered several task forces to pursue the following goals:

- Define required body of knowledge and recommended practices.
- Define ethical standards.
- Define educational curricula.

Additional activities will include coordinating the work of the steering committee other with CS and ACM boards and committees, other professional societies, and other communities.

E-8A Organizational Certification of Software Engineers (feasibility study)

Because software engineering is still an emerging discipline, there are no widely accepted definitions of the body of knowledge and skills that a software engineer should possess, and no mechanisms for certifying that a software engineer possesses that knowledge and those skills.

Over the past six months, the SEI has received many requests from industry for information on certification of software professionals. The contexts for these requests have been varied:

- What to look for in new hires from universities, or what to look for in a university curriculum.
- What to look for in hiring experienced personnel.
- Who to assign to a specific project.
- Standards for promotion or salary increases.

In addition, we have received requests for guidance on what constitutes appropriate continuing education or re-education for experienced engineers.

Objectives

We propose developing a mechanism that can be used by the software community to satisfy its certification needs. Such a mechanism could establish a standard for minimal competency that is higher than what is now found in industry. If the mechanism is widely accepted, it would provide motivation for individuals to improve their knowledge and skills, and guidance to organizations on improving their software capabilities by improving the knowledge and skills of their

software engineers. This is one specific mechanism that might appear in a larger framework such as the People Management Capability Maturity Model (PMCMM) (see page 60 for details).

In addition to helping large software organizations, the objectives of this work are to provide:

- A basis for designing academic curricula.
- A basis for developing accreditation guidelines for academic programs.
- A standard for third-party education and training vendors.
- A guideline for defining career paths in military software.
- A basis for improved certification standards of the Institute for Certification of Computing Professionals.
- A basis for potential licensing examinations (perhaps through the National Council of Engineering Examiners) if and when licensing requirements for software engineers are adopted by the states.

Proposed Work

The proposed work consists of three phases, beginning in 1995 and continuing through 1997:

- *Problem definition and feasibility study.* This phase requires the participation of representatives of several of the SEI strategic partners and other customers. It would examine the needs of government and industry with respect to in-house professional certification of software engineers. It would determine the feasibility of the SEI developing a certification model (meaning definitions of the knowledge and skills required of software engineers at various career phases) that could then be tailored and instantiated within a particular company.

- *Identification and organization of the body of knowledge.* This phase would build on the results of the feasibility study, on past SEI curriculum design efforts, and on the IEEE Computer Society's 1994 effort to identify the body of knowledge that industry agrees constitutes software engineering. With continued participation from SEI strategic partners and customers, the body of knowledge would be elaborated and organized in ways appropriate for establishing certification criteria.

- *Development of assessment instruments.* This phase would develop generic written examinations and other instruments for assessing the knowledge and skills of software engineers. The instruments would be prototyped with the participation of SEI strategic partners and customers, who would also collect data to improve and validate the instruments. Instructions for tailoring the instruments to particular industries or companies would be developed.

Note: We are not proposing that the SEI certify software engineers. Nor are we proposing some kind of national certification mechanism. The proposed work is to develop a certification capability that individual government and commercial organizations can tailor and use within their own organizations.

For 1995, the work would include the first phase only: problem definition and feasibility study. If the work is judged feasible, a detailed plan would be proposed for work to continue in 1996 and 1997.

4.3 Services in Technology Transition

Software engineering organizations need advice and guidance on identifying and implementing software engineering improvements. The SEI responds by developing, delivering, and transitioning services that help SEI customers improve their ability to design, develop, maintain, and operate software-intensive systems. To accelerate the widespread adoption of effective software practices, we:

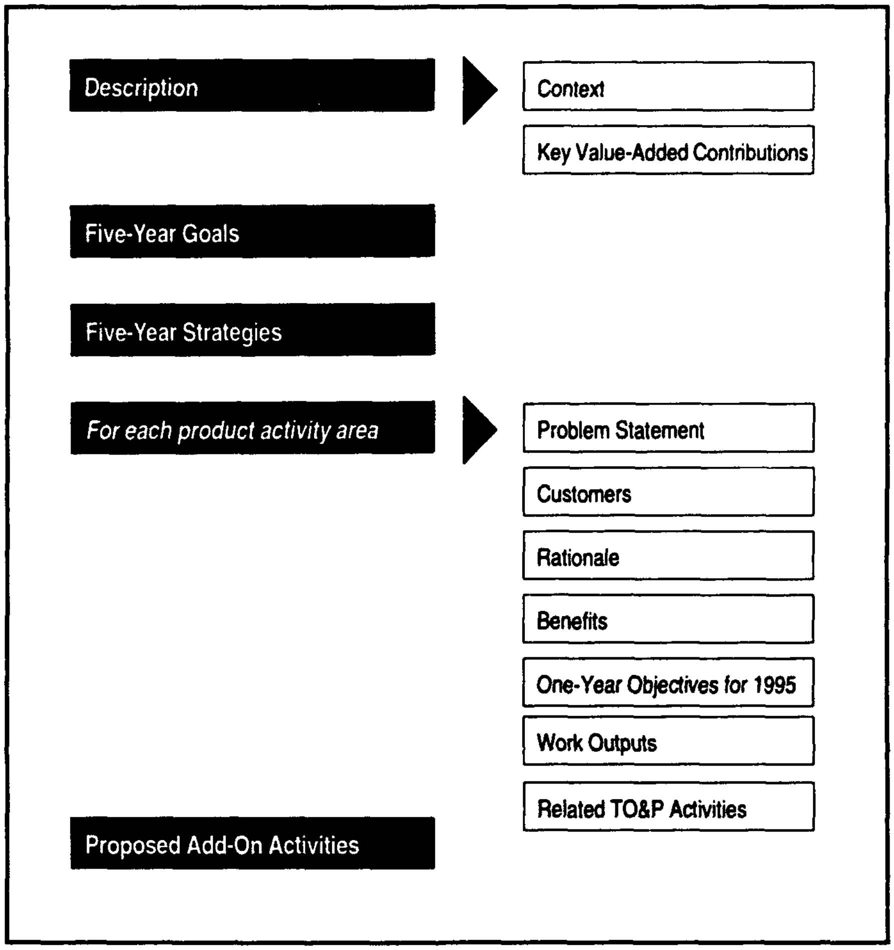
- Work with organizations that are influential leaders in the software community.
- Help those organizations leverage their resources by aligning their software engineering improvement activities with other organizational activities.
- Promote the development of infrastructures that support the adoption of improved practices.
- Transition capabilities to government and commercial partners for use with their customer organizations.

Our role is to demonstrate the applicability and utility of software engineering technology by providing direct support and guidance to software engineering organizations in different applications domains. Our aim is to help them build their internal capacity to initiate and sustain improvements in their software development and maintenance activities and in the processes they use to adopt and institutionalize new technologies. Our infrastructure-building activities allow managers and practitioners to share lessons learned with a broader community to motivate other organizations to invest in improvement. These customer-focused efforts demonstrate the benefits of investments in software engineering process and technology.

Recognizing the need to influence a broad community and the need develop a better understanding of the cultures and needs of different types of organizations, the SEI provides services to a broad range of clients including those listed below:

- Electronic Systems Center (ESC)
- Embedded Computer Resources Support Improvement Program (ESIP)
- Software Technology for Adaptable, Reliable Systems (STARS)
- Software Technology Support Center (STSC)
- Air Force Materiel Command (AFMC)
- Army Materiel Command (AMC)
- Naval Oceanic Office (NAVOCEANO)
- Marine Corps Tactical Systems Support Agency (MCTSSA)
- Sandia National Labs
- Union Switch and Signal
- U.S. Treasury
- Xerox
- US West
- Federal Express
- Air Force Space Command (AFSPACECOM)
- Air Force Program Executive Office for Management Information Systems [PEO (MIS)]
- Space and Naval Warfare Systems Command (SPAWAR)
- National Oceanic and Atmospheric Administration (NOAA)

The sections for this area of expertise are:



4.3.1 Description

4.3.1.1 Context

Our approach to the development and delivery of services is client centered. We work with client organizations to identify their software engineering improvement needs and to develop and implement improvement plans that allow the clients to satisfy those needs. To help the clients implement their plans, we draw products and expertise from all areas of the SEI as well as from other selected technology development activities (e.g., ARPA STARS program). Customer teams are formed to bring the outputs of the focus areas, the SEI core competencies, and the technologies available from other producers together to focus on the clients' improvement goals.

Since an organization must learn to manage its software processes before it is likely to see large benefits from the adoption of new technologies, our focus has been on helping client organizations build and sustain continuous software process improvement. Now, as these orga-

nizations assess their needs and implement their improvement plans, they recognize the need to integrate software engineering processes, methods, and tools with each other and with other organizational goals, strategies, and systems. The SEI is responding to these needs by broadening the scope of its service activities. New activities include developing, testing, and transitioning processes and methods that enable software engineering managers and change agents to address their process, technology, people skills, and organizational problems in a way that recognizes the interaction of the improvement activities and maintains a focus on the organization's overall goals and strategies.

4.3.1.2 Key Value-Added Contributions

SEI-led efforts at AFMC and AMC demonstrate the benefit of SEI services and technology transition assistance. These two organizations represent 12 separate software development and maintenance centers. Each center has been assessed and has developed improvement infrastructures and plans. All 12 are making progress toward their goals of improved software capability—several have achieved the next maturity level.

The SEI has fostered the growth of participation in the SEPG National Meeting from 46 attendees in 1988 to over 700 in 1994. This growth has been accomplished primarily through working collaboratively with SPINs throughout the U.S. The SEPG National Meeting now has the infrastructure to accept and promote other software engineering and improvement techniques. The SEPG National Meeting has the following objectives:

- Provide information on initiating and sustaining SEPG activities.
- Advance the state of SEPGs.
- Establish a network mechanism for SEPGs.
- Enable those involved in software process improvement to discuss issues and learn from each other.

As a result of the services provided to AFMC, AMC, and others, the software community has developed an understanding of the criticality of organizational and personnel variables to its success in improving. SEPG members, their sponsors, and other change agents are now expected to have skills and knowledge in managing change, dealing with resistance, continually assessing readiness, and adjusting their improvement strategies and plans accordingly. This understanding is directly traceable to the work done by the SEI in its support of client improvements.

4.3.2 Five-Year Goals

Our goal is to foster improvement in software practice by working directly with software engineering organizations and helping them institutionalize continuous process improvement and adoption of software engineering technology. We plan to support our customers' improvement efforts so that SEI customers will recognize the benefits of investments in software engineering improvements, will be operating with basic software management processes in place, and many will be using standardized, integrated, organization-wide software processes. Because

of this, by the year 2000 our customers will have in place standard planning and organizing processes for improvement that they can apply to the adoption of new technology. Many of these organizations will have developed the capacity for long-term continuous improvement and will have organizational infrastructures in place to facilitate ongoing improvement efforts.

In addition, it is our goal to foster the continued evolution of infrastructures within and across organizations so that, by 2000, the software profession has a self-sustaining infrastructure that supports organizations' efforts to improve their practice in a measurable way. This infrastructure consists primarily of existing delivery systems and professional and trade groups that organizations already use. Improving software engineering practice will be the focus of a major national conference. Government, industry, and trade groups that depend on software will be represented at the national conference and the majority will include software engineering improvement sessions and tracks on the agendas of their conferences.

Within five years, the benefits of organized, integrated software engineering improvement will be recognized to the point that there will be a market for government and industry service providers to extend their service offerings to include software engineering improvement services and to offer products and services licensed from the SEI.

4.3.3 Five-Year Strategies

To achieve these goals, our strategy is to focus on, develop, field test, and transition a structured approach to software engineering improvement. This activity will give software engineering managers guidance in integrating their business and organizational improvement goals with technology improvement, process improvement, and skill development. It will also provide service providers and change agents with a process they can use when supporting improvement activities at their client sites. In more detail, our strategy is to:

- Build on earlier process improvement work and continue developing and providing a structured approach to software engineering improvement.
- Broaden the impact of SEI work by selecting client organizations from a broad cross-section of software-intensive application domains: i.e., DoD, other government agencies, and industry segments. We will work with these clients through TO&P, technical collaboration agreements (TCAs), and cooperative research and development agreements (CRADAs).
- Continually enhance the improvement approach by working with a variety of technology producers to demonstrate the benefit of their technologies and incorporating references to those technologies into the improvement approach.
- Provide direct support to software organizations that will enable them to develop their internal capacity for (1) identifying their improvement goals, (2) developing and implementing plans to achieve those goals, (3) managing the organizational and personnel changes associated with these improvement efforts, and (4) creating an infrastructure and strategies for ongoing adoption and refinement of processes and technologies.
- Sponsor the annual SEPG National Meeting. Work with government, industry, and trade associations to include software engineering improvement topics on the agendas of their conferences.

- Transfer to the software community the capability to deliver the training and services we currently provide.

4.3.4 Software Engineering Improvement

This section discusses the product activity area related to software engineering improvement.

4.3.4.1 Problem Statement

TO&P clients and industrial contacts seek out SEI assistance, but are often uncertain about what problem they are trying to solve or what outcomes they are trying to achieve. Managers in client organizations are struggling to deal with changes in their environments—changes that are often dramatic and occur at an increasing rate. At the same time organizations are dealing with changes in their environments, they also face rapidly changing technology. Managers are struggling to deal with these changes and the technical and cultural shifts that accompany them.

Over half of our clients are insisting on more than point solutions to their technical problems or improvement efforts focused only on process. As shown in Figure 4-10, these clients must simultaneously work on technology, process, human resources, and organizational problems. They recognize that improvements in any area can be sustained only when the improvement activities are planned and conducted in the context of the overall organization, where the interaction of the organization's various components (e.g. people and skills, culture, systems, strategy, goals and values, work processes, and structure) are considered.

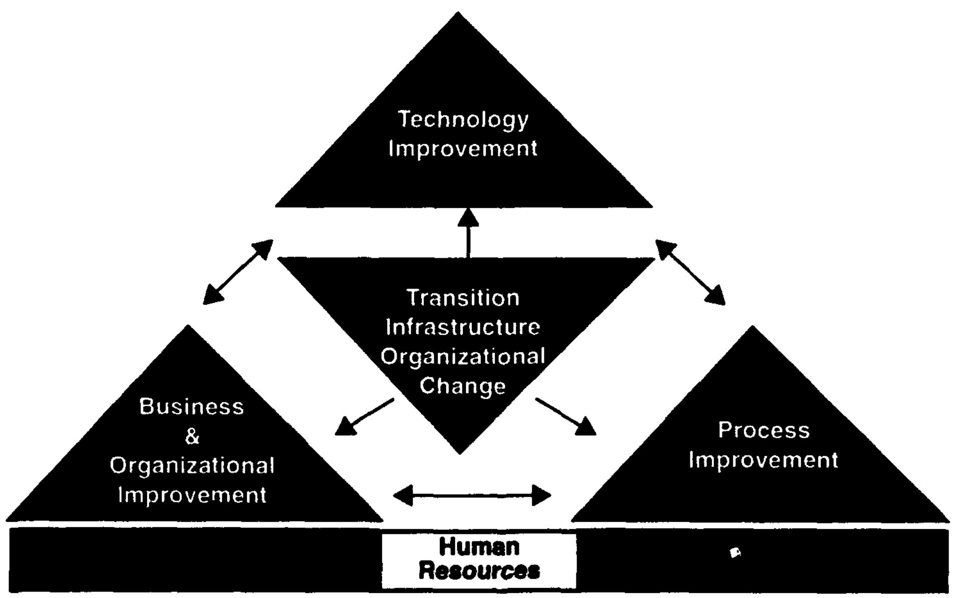


Figure 4-10: Coordinated Improvement Activities

4.3.4.2 Customers

Activity in this area is focused on producing processes and methods that will help two classes of clients:

1. Software engineering managers and practitioners who are trying to define, plan, and implement processes, methods, and tools that lead to sustained improvement in their software engineering capability.
2. Change agents, consultants, and other service providers who support these change efforts.

4.3.4.3 Rationale

The SEI has been active at client sites helping them implement changes for over eight years. In addition to benefiting a particular client, many of these efforts have helped the SEI prove concepts, develop methods, and develop products. More recently, the SEI has broadened these improvement activities to integrate much of our work at the clients' sites to help the clients establish and sustain continuous improvements—both process improvements and adoption of new technologies.

Figure 4-10 illustrates the close interdependencies between technology improvement, organizational change, business and organizational improvement, process improvement, and human resources skills. From our field work, we find that our clients are often very good at analyzing their own situations and at identifying their own strengths and weaknesses, but are not as effective at developing and executing realistic improvement plans. This is often a consequence of the failure to recognize the interdependencies pictured in the figure. Our clients often overlook the organizational aspects of change and fail to do things like secure management sponsorship and leadership for change efforts, assess the organization's readiness for change, develop effective communication channels, or adjust reward systems to reward new behavior rather than old behavior. In addition, because change activities are often defined without regard for the overall organization's goals and strategies, local change efforts attempt to move an organization in one direction while strategically it is trying to go in another.

Our clients often do not know how to take the crucial step from organizational motivation, goal setting, and preparation of the organizational infrastructure for accepting changes to putting improvements into effect at the project level. It is difficult for them to translate broad organizational goals into specific project objectives such that needed software engineering practices can be tried and instituted on projects on a "just-in-time" basis. Furthermore, there is often no formal mechanism for measuring and propagating the successes and improvement lessons learned from one project into other projects within the same organization, or for reflecting progress at these project-level "grass roots" back into software engineering improvement milestones established at the organization level.

As a result, improvement efforts often either fail outright or suboptimize their results: the new technologies or processes are not used or are not supported strategically and in day-to-day decisions. Change agents, working within organizations to bring about improvements, often

show strong understanding of particular technologies or particular processes, but have less understanding of technology transition and organizational issues, especially the need to identify the dependencies and manage the interfaces across the various activities.

As we began educating the community about these issues, we began to see a dramatic increase in demand for assistance in developing improvement plans that integrate process, technology, skill development, and organizational activities.

4.3.4.4 Benefits

Figure 4-11 depicts the software engineering improvement trends over the next four years.

Raising awareness of the need to integrate software engineering improvement with other organizational activities and providing a structured method for this integration gives the community a path to follow that naturally extends their process improvement work. As software organizations are better able to relate their process improvement work to technology adoption, skills development, and other organizational activities, they will produce improvement plans that have a higher probability of success. These plans will describe the relationship between the improvement activities and other organizational goals and objectives. Expected results of these activities will appear as measurable objectives for senior management to help insure that the activities receive the attention and sponsorship they need for success.

Organizations will have a structured approach to use in identifying and planning their improvement activities and will be better able to determine the risk of process improvement or new technology adoption before undertaking the effort. In addition, organizations using this approach will have ways to gauge specific areas of resistance to change and other risks. They will have techniques to deal with these problems and will be better able to manage the change efforts. Improvement activities will be better aligned with other organizational systems, people, strategies, and culture, and improvement goals will be achieved more efficiently and effectively. These aligned organizations will have a better chance of managing continuous improvement and technology transition and will see a better return on their improvement investments.

A structured, documented improvement method will allow the SEI to transition its expertise to change agents within software engineering organizations as well as to commercial service providers. Today, many change agents focus on specific technologies or process improvement activities. Organizations working on multiple problems must bring together different people with different areas of expertise and must somehow integrate these activities themselves. As change agents broaden the scope of their work and add new products and services to their offerings, their customers will benefit by reducing the number of different people involved in change activities and by reducing their integration efforts. These benefits will, in turn, create a market for integrated services and serve to give SEI work a greater impact.

As more organizations adopt structured improvement methods and are better able to identify and manage the risks associated with the adoption of new technologies, they will be in a better position to experiment with and adopt newer, promising technologies. Technology producers

will have more opportunities for demonstration projects and will receive better feedback on the utility of their technologies and the ability of organizations to adopt them.

Key Items	State of practice as of:			Impact/Metrics
	Today	+2 Years	+4 Years	
Improvement plans integrate process, technology, skills development, and organizational issues.	Efforts focused on single changes (e.g. process, particular technology) with little attention to related activities on organizational dependencies.	Plans begin to link process, technology, skills, organizational issues.	Plans show interrelationship and dependencies between process; technology adoption; skills development; organizational vision, strategy, goals, objectives, and subsystems.	Expected results of software engineering improvement activities appear as goals and measurable objectives in senior management annual plans.
Organizational tools used in planning and implementing improvement efforts.	Limited use in process improvement efforts—some faltering use in technology adoption.	Widely used in process improvement, more consistent use in technology adoption.	Widespread use with some clear impacts.	Increased interest in classes and requests for help; increased use of organizational data apparent in process improvement and technology adoption plans.
Technology producers demonstrate utility.	Limited availability of test sites able to identify specific reasons for level of acceptance of technology.	Early adopters can identify risks of adoption, develop improvement plans, and develop risk mitigation plans.	Early adopters can pinpoint why new technology was or was not successful.	Increased availability of test sites; increased number of adoption plans with risk mitigation strategies.
Change agents use integrated approaches.	Internal change agents and consultants focus on specific technologies or on process improvement services.	Change agent product and services offerings incorporate wider range of technology adoption, process improvement, and skills development activities. SEI transition partners use SEI improvement framework with their customers.	Change agents offering integrated products and services have clear market advantage over those offering point solutions.	Customers express increased satisfaction with improvement services. Each organization's vendor of choice offers integrated services.
Integrated improvement concepts adopted by broad SE community.	Conference presentations and lessons learned reports focus on point solutions.	SEPG National Meeting and other major conferences feature tutorials on integrated approaches.	SPINs and SEPGs broaden activity to include technology adoption, skills development, and organizational development activities.	Integrated software engineering improvement efforts appear on the programs of software conferences and engineering workshops.

Figure 4-11: Trends in Software Engineering Improvement

Overall, the software engineering community will gain an appreciation for, and benefit from, a more holistic approach to software engineering improvement. The search for silver bullets will be replaced by structured efforts that treat improvement activities as ongoing projects that integrate with other organizational activities. Early investments made in process improvement activities, including infrastructures such as SEPGs and SPINs, will serve as a foundation for improvement activities in areas beyond process. SPIN and SEPG members will report increased use of integrated efforts, along with increased satisfaction in the results of these efforts. As a result, the infrastructures established to support process improvement will evolve to support the adoption and integration of all technologies and techniques that bring benefit to software engineering organizations.

4.3.4.5 One-Year Objectives for 1995

During 1995, the software engineering improvement effort will:

- Coordinate with the SEI focus areas to bring together their work to refine and field test a comprehensive "software engineering improvement method" consisting of an overall improvement approach and a set of processes and methods that managers and change agents can use to identify, plan, and manage their improvement activities.
- Present the improvement approach and initial lessons learned at key conferences in the form of papers and tutorials.
- Draft training material that can be used to train change agents and other service providers in 1996; pilot this material with in-house customer support teams.
- Identify technology development activities that show high promise for early adoption at improvement client sites.

4.3.4.6 Work Outputs

During 1995, pilot versions of the outputs described below will be available to TO&P and industry clients. After additional work in 1996, these outputs will be released to the community.

Software Engineering Improvement Framework (1995-1996). This output documents an approach to software engineering improvement from two points of view: the change agent view and the software engineering organization (customer) view. The change agent view is based on the eight-phase consulting model that the SEI has designed to help SEPG members work more effectively in their organizations. This model will be elaborated to cover the integration of technology, process, skills development, and organization development activities including:

- Processes and techniques multiple change agents will use when working together to deliver improvement services.
- Decision and conflict resolution processes.
- Roles and responsibilities definitions.
- Feedback processes to capture lessons learned and adjust future improvement activities.

The customer view documents a vision of continuous software engineering improvement. It provides a life-cycle model for software engineering improvement and a generic roadmap for improvement activities that describes the overall process, resources, and management structures required to define, plan, manage, and conduct the improvement efforts.

Tailoring guidelines (1995-1996). This output provides guidelines that individual organizations can use to tailor the framework to create the organization's own applied software engineering improvement framework. The applied framework includes the organization's software engineering vision, the organization's improvement life-cycle model, and the organization's improvement roadmap. Organization-specific versions of the framework are needed to provide the context for ongoing, continuous improvements and to help insure consistency of activities over time.

Draft training material for software engineering improvement (1995-1996). As the Software Engineering Improvement Framework is developed and piloted with TO&P customers, training material that can be used to train change agents will be developed. This material will initially be piloted with SEI staff in 1995 to test the material and build SEI competence in using and training people in the improvement approach. The material will then be used as the basis for a course to train external change agents and other service providers in 1996.

Additional materials generated during 1995 from this project will include special reports on interim experiences in developing and testing the software engineering improvement method and conference presentations to make visible our intent to transition an integrated improvement method to the community.

4.3.4.7 Related TO&P Activities

The work outputs listed above will be developed incrementally through the application of early versions of the work to the improvement efforts of TO&P customers. For example, improved methods for profiling and prescreening customers' current state of the practice and readiness for specific improvement activities can be validated on existing and proposed TO&P customers. In addition, existing SEI methods, guidelines, and other outputs will be integrated at TO&P customers' sites through initial versions of the framework.

4.3.5 Proposed Add-On Activities

There are no proposed add-on activities in this area.

4.4 Customer Involvement

The SEI has developed a range of relationships, from those that satisfy customer needs for general information about the SEI and its technical program to those that involve collaboration and technology transition. The SEI provides opportunities for customers to participate in workshops, conferences, advisory boards, and educational offerings as well as the acquisition and co-development of specific products and services including customer-site support. The ways in which our customers can work with the SEI are described below.

4.4.1 Customer Inquiry/Response

This service is provided through the SEI information line: (412) 268-5800; Internet email: `customer-relations@sei.cmu.edu`; and FAX: (412) 268-5758. The customer information specialists who provide this service are fully prepared to answer any question of a general nature about the SEI, to mail pertinent descriptive materials, and to follow up with members of the SEI technical staff to provide more detailed information. The SEI provides this service during normal working hours, handling up to 250 requests per week. We collect statistics and maintain a database reflecting the character of the inquiry/response traffic. This information is often used by others at the SEI to contact and respond to our customer community.

Another way of reaching customers is through Visitor's Day, which is hosted by the SEI three times a year to familiarize software managers, practitioners, and educators with the SEI and its activities. Members of the SEI technical staff, program managers, and project leaders give presentations on the technical program and SEI products and services. Preview demonstrations of SEI technical offerings are also frequently showcased.

4.4.2 Subscriber Program

The Subscriber Program is an effective way for an individual to stay informed about SEI activities. Participants receive mailings that keep them up to date on SEI events, course offerings, work in progress, new products, and new initiatives. Anyone with a United States mailing address is eligible to subscribe. A fee of $100 (as of January 1994) has been established to help offset costs of delivery. The fee covers an entire year from the date that the subscription is activated. It applies to industry and academia; government customers receive the same benefits at no cost by controlled distribution. Major upgrades to the Subscriber Program are under consideration for 1995.

SEI subscribers currently receive the following:

- The quarterly *Bridge* magazine.
- The annual *Technical Review.*
- Discounts on technical reports.
- A substantial discount at the annual SEI Software Engineering Symposium.
- Early notification of SEI events.

4.4.3 Resident Affiliate Program

The Resident Affiliate Program provides the opportunity for experienced technical personnel from government, industry, and academic organizations to participate in SEI projects. Resident affiliates contribute both as software engineers and as application domain experts, providing a valuable and practical perspective. They help us understand our customers' needs by providing information about their home organizations that helps us understand the organizational and technical contexts in which they practice software development.

The sponsoring organizations benefit by participating in technical activities that might not be possible in their own organizations, by obtaining early the results of SEI technical activities, and by having access to SEI people, projects, and other resident affiliates. The resident affiliate benefits from working in a different technical context, from participating in the many workshops and other activities at the SEI and in the larger CMU community, and from interacting with colleagues from different professional, technical, and organizational backgrounds. The SEI also benefits because it obtains experience, expertise, and additional insight into the software engineering community.

Resident affiliates work on-site at the SEI for a negotiated period, usually 6 to 24 months; they may spend half- to full-time at the SEI. They devote approximately 80 percent of their time to an SEI technical project and the remaining time to technology transfer and liaison activities with their home organizations. Resident affiliates are treated as integral members of the SEI staff.

Work outputs for 1995 include:

- Coordination of the Resident Affiliate Program.
- Resident affiliate attendance at SEI course (Managing Technological Change and Consulting Skills Workshop).
- Maintenance of a resident affiliate handbook, directory, and database.
- Monthly information exchange meetings.

Organizations that have participated in the Resident Affiliates Program are listed in Appendix C.

4.4.4 Advisory Boards/Working Groups

From time to time, the SEI identifies the need for a customer advisory board or working group to provide customer guidance on current activities and future plans and to perform technical reviews of products. Members are selected through a screening process using project-defined criteria intended to populate the board or group with a mix of technical professionals who can help satisfy SEI technical objectives. The current advisory boards and working groups include the following:

- Software Process Advisory Board
- Software Process Definition Advisory Group
- Software Process Measurement Steering Committee

- Software Process Assessment (SPA) Vendors Association
- Systems Engineering CMM (SECMM) Steering Group
- Capability Maturity Model (CMM) Advisory Board
- CMM-Based Appraisal (CBA) Advisory Board
- Software Capability Evaluation (SCE) Review Group
- People Management Capability Maturity Model (PMCMM) Advisory Board
- Software Dependability Working Group
- Educational Products Advisory Board
- CERT Advisory Task Force

For a description of each of these groups, see Appendix B.

4.4.5 Value of the SEI Study

The Value of the SEI effort is a multi-year study to identify and quantify the impact that SEI products and services have on software engineering practices, software organizations, and the software community as a whole. SEI products and services addressing process, technology, and education will be analyzed. Data on the customers, use, and impact of SEI products and services will be gathered, stored in a database, and analyzed. Work in 1995 will include the development of a plan and pilot studies. Audiences for this work include SEI sponsors, product and service developers, and customers.

4.4.6 Software Process Improvement Network (SPIN)

In September 1992, the SEI agreed to serve as coordinator for the emerging SPIN. The purpose of the SPIN groups is to satisfy customer needs for a practical forum for exchanging ideas, information, and mutual support on the subject of software process improvement. The primary role of the SEI is to disseminate information from existing SPIN organizations to groups of people in common geographical locations who are interested in starting new SPIN groups. The SEI maintains a directory of all currently active SPINs and points of contact in areas where interest has been expressed in forming a new SPIN.

Active SPIN organizations have been formed in Washington, D.C.; Irvine, Calif.; Dallas, Texas; Austin, Texas; Boston, Mass.; Seattle, Wash.; Boulder, Colo.; St. Louis, Mo.; San Ramon, Calif.; Northern New Jersey, Omaha, Neb.; Northeast Ohio; Huntsville, Ala.; Phoenix, Ariz.; Los Angeles, Calif.; Albuquerque, N.M.; and Tucson, Ariz.

International locations include the United Kingdom; Montreal, Canada; France; Spain; and Bangalore, India.

New SPINs are being initiated in Pittsburgh, Pa.; Houston, Texas; Silicon Valley, CA; Salt Lake City, Utah; Hampton Roads, Va.; Milwaukee, Wis.; and Southwestern Ohio.

The SEI also maintains an email alias used to disseminate announcements of interest to the network and distributes SPIN start-up information on forming SPIN organizations.

The SPIN provides significant leverage for transition. The SEI interacts with SPIN groups, each member of which represents one or more SEPG. Each SEPG, in turn, represents a corporate software engineering staff often measured in the hundreds. Figure 4-12 illustrates this one-to-many effect.

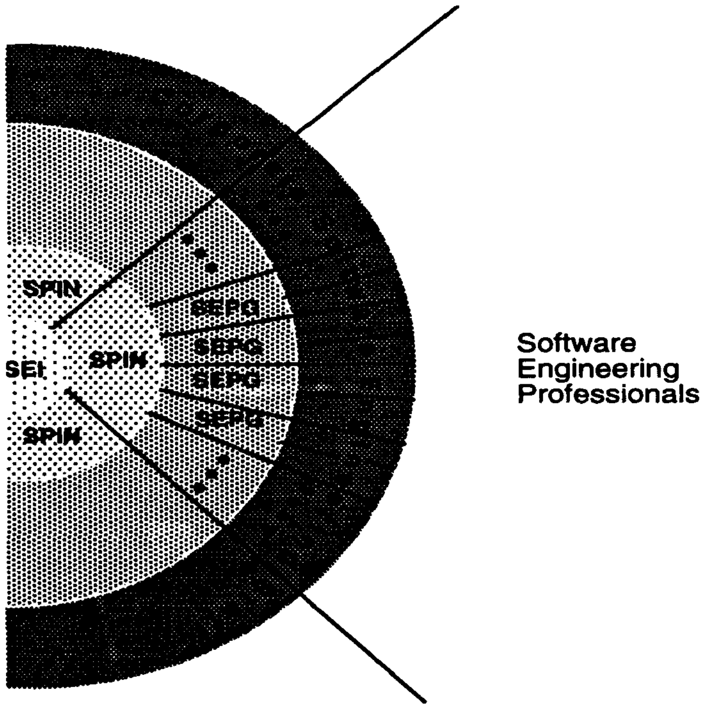


Figure 4-12: Effect of Software Process Improvement Networks (SPINs)

In 1995, existing work will be continued: the *SPIN Directory* and the SPIN start-up information will be maintained and updated, and a newsletter will be written and distributed on a regular basis. Currently information is distributed electronically or otherwise at least weekly.

4.4.7 Collaboration Programs

SEI collaboration programs are intended to create well-defined and well-managed relationships with the community of industry customers. Through these partnerships, the SEI has access to an industry constituency that can:

- Provide input to the SEI technical program.
- Advance the maturity and accelerate the development of SEI technology, products, or services.
- Provide in-kind and direct funding resources.

4.4.7.1 Technical Collaboration Program

Technical collaborations are formed for a fixed duration, involving well-defined areas of interest with one or more SEI technical projects, with the end objective of a demonstrable result. Current examples include co-development of products of mutual interest (for example, roadmaps, field guides, handbooks, and training courses), technology exploration, and pilot/field testing of new products or processes. Technical collaborations are initiated by mutual agreement and are negotiated between the project and the potential partner with the intent of exchanging value of mutual benefit. Organizations that have participated in our technical collaboration program are listed in Appendix D.

4.4.7.2 Strategic Collaboration Program

Strategic collaborations are long-term, corporate-level relationships between the SEI and selected industry partners. These partnerships are characterized by mutual statements of strategic intent and goals. The strategic relationship is realized by executing multiple technical collaboration agreements, as described above.

Candidate strategic partners have demonstrated their commitment to the SEI mission and vision and the transition of SEI-developed approaches by virtue of their historical and current involvement with the SEI. In addition, they have demonstrated a strong commitment to continuous improvement in the quality of their own software products and processes. The SEI seeks partners who:

- Are recognized leaders in several market segments.
- Have an ability to execute technology transition roles.
- Can contribute to the depth and breadth of the SEI technical program.

Benefits for strategic collaboration partners include:

- Broader, and often more immediate, access to SEI products, services, and technical staff.
- An opportunity to have input into the SEI technical program.
- Early access to products being co-developed, including products that may have been difficult to develop in a timely way without the benefit of collaboration.

Current strategic collaboration partners include Hewlett-Packard, Hughes, Loral Federal Systems (previously IBM Federal Systems Company), and Texas Instruments.

4.4.7.3 Cooperative Research and Development Program

As part of the collaboration program, the SEI is now able to engage in CRADAs with industry organizations. The Federal Technology Transfer Act of 1980 and the National Competitiveness Technology Transfer Act of 1989 give organizations such as the SEI latitude to enter into these agreements. The intent is to accelerate the transition and commercialization of technologies by permitting the SEI to receive funds from industrial organizations that have a commercial interest in technology work in progress. This program is gaining momentum and will contribute significantly to the SEI technical program in 1995.

4.4.8 SEI Software Engineering Symposium

This symposium is a major SEI event and perhaps our most visible technology transition activity. Held annually, it is the primary forum for presenting SEI work to the U.S. software engineering community. The symposium program includes a significant number of presentations from industry and government customers whose technology results relate to SEI work in progress.

This year's theme—The SEI Celebrates 10 Years of Improving the Practice of Software Engineering—reflects the milestone of the Institute's tenth year of operation. More than a reflection of past SEI software engineering accomplishments, this year's event will include major technology transition efforts within the institute and in the software engineering community. This year we are mirroring the current *1&5 Year Plan* in terms of communicating the strategic framework of the SEI and core competencies for improving software practice. The three tracks integrate work in maturing the technology, maturing the process, and maturing the profession. Maturing the profession focuses on software technology transition efforts. An increase in outside invited speakers from industry and government ensures balanced, timely information for the entire software engineering community. The symposium provides a valuable opportunity to our customers and ourselves, not only to learn about mutually beneficial activities but also to interact, to learn of emerging software engineering technology applications, and to provide feedback to one another. It is one of our largest and most successful technology transition events, given that people are still the most effective means for technology transition.

Since the inception of the symposium in 1987, attendance at the event has increased from 360 to nearly 1300. Figure 4-13 shows the growth in attendance over the past five years. More importantly, an increasing number of presentations are given by our customers —approximately one-third in 1993 and 40% in 1994. In 1995, our goal is to have half the presentations at the symposium given by our customers.

Many other events are held for customers who are interested in specific aspects of software engineering or SEI work; for example, the Risk Conference (see page 78), and the Conference on Software Engineering Education (see page 168). These and other events are in our product lists and are described in various sections of this document.

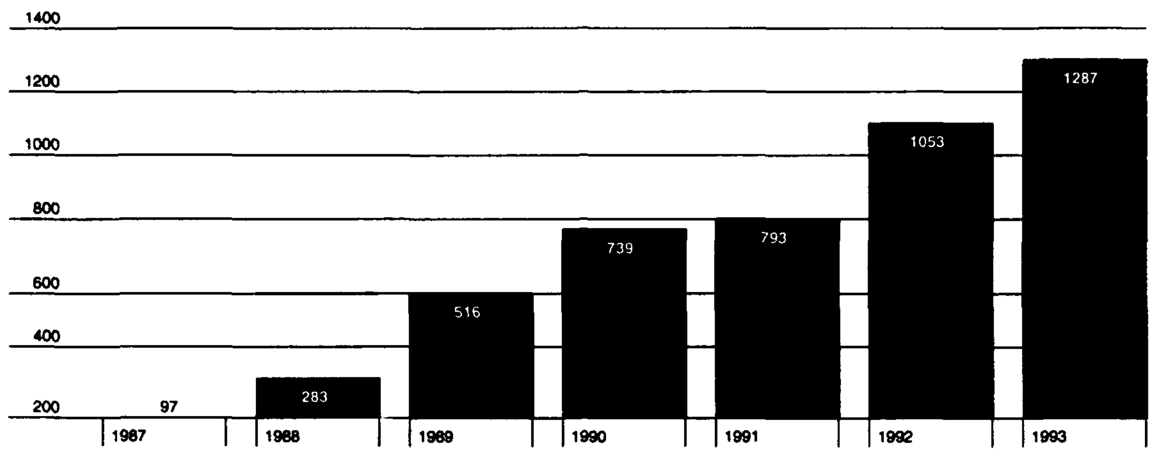


Figure 4-13: Attendees at SEI Symposia

4.4.9 Software Engineering Process Group (SEPG) National Meeting

The objectives of the SEPG National Meeting are to:

- Provide information on initiating and sustaining SEPG activities.
- Advance the state of SEPGs.
- Establish a network mechanism for SEPGs.
- Enable those involved in software process improvement to discuss issues and learn from each other.

Attendance at this event has grown significantly over the past six years—from 46 in 1988 to over 700 in 1994.

4.4.10 Proposed Add-On Activities

The following section describes a proposed core add-on activity in customer involvement. The SEI is asking selected members of the software community to evaluate the relative attractiveness of these add-on proposals as possible elements of the final 1995 core program, as funding permits. Appendix A lists all baseline and add-on items.

CI-1A Customer Relations

The Customer Inquiry/Response Program at the SEI provides for broad-based information dissemination about the SEI technology program, products, services, and collaboration opportunities. These services directly support the technology transition mission of the SEI.

Customer Inquiry/Response is provided through the SEI information phone line, Internet email, and FAX. Customer information specialists are fully prepared to answer any question of a general nature about the SEI, to mail pertinent descriptive materials, and to follow up with

members of the SEI technical staff to provide more detailed information. The SEI provides this service during normal working hours, handling up to 250 requests per week and off-loading these information requests from all projects and programs. We collect statistics and maintain a database reflecting the character of the inquiry/response traffic. This information is often used by others at the SEI to contact and respond to our customer community.

Appendix A List of Baseline and Add-On Items

Item Designator	Item Name	Core: Baseline	Core: Add-On	Proposed TCSP	Dependencies	Page
SP-1B	Community Involvement	■			SP-3B	42
SP-2B	ISO SPICE Technical Reports	■			SP-1B, SP-3B, SP-6B	48
SP-3B	CMM Revision and Maintenance	■			SP-1B, SP-2B, SP-6B, SP-1A	48
SP-4B	Software Process Definition Guidelines	■		■	SP-1B, SP-3B	52
SP-5B	Personal/Team Software Process Research	■			SP-1B	52
SP-6B	CMM Validity Studies	■			SP-1B, SP-7B	57
SP-7B	SEI Process Database	■			SP-1B	58
SP-8B	Appraisal Methods	■			SP-1B, SP-3B, SP-10B	58
SP-9B	Value of the SEI	■			SP-1B, SP-6B, SP-7B	58
SP-10B	Appraisal Architecture and CRF Report	■			SP-3B, SP-8B	59
SP-1A	Systems Engineering CMM (SECMM) Development and Integration		■		SP-1B, SP-3B, SP-3A	59
SP-2A	People Management Capability Maturity Model (PMCMM)		■	■	SP-1B, SP-3B	60
SP-3A	Systems Engineering Capability Maturity Model (SECMM) Maintenance		■		SP-1B	60
SP-4A	Process Research Program		■		SP-1B, SP-5B	61
SP-5A	Software Process Definition Framework		■	■	SP-1B, SP-3B	61
SP-6A	Process Value Method		■		SP-1B, SP-6B, SP-7B	61
SP-7A	Software Measurement Handbook		■		SP-1B, SP-3B	62
SP-8A	Software Product Measurement Definitions		■		SP-1B, SP-3B	62
SP-9A	Empirical Methods Product Improvement		■		SP-3B, SP-8B	63

Figure A-1: Baseline and Add-On Items for the Software Process Focus Area

Item Designator	Item Name	Core		Projected TCSP	Dependencies	Page
		Baseline	Add-on			
RM-1B	Risk Action Planning Guidebook	■				78
RM-2B	Tailorable Taxonomy-Based Questionnaire	■		■		78
RM-3B	Risk Repository	■		■		79
RM-4B	Predictive Decision Tool	■				80
RM-5B	Program Manager's Assistant	■			RM-3B	80
RM-1A	Technology Alignment Guidelines		■			82
RM-2A	Software Risk Assessment for Manufacturing		■			82
RM-3A	Cost Model Risk Management Method		■	■		82
RM-4A	Technology Assessment Taxonomy-Based Questionnaire		■			83
RM-5A	SRE Train-the-Trainer Course		■			83
RM-6A	State-of-the-Practice Report		■			84
RM-7A	Risk Management Key Practice		■	■	SP-1A, SP-3A	84
RM-8A	Acquisition Risk Management Guidelines		■			84
RM-9A	Software Acquisition Capability Maturity Model (SACMM)		■	■	SP-1A	84

Figure A-2: Baseline and Add-On Items for the Risk Management Focus Area

Item Designator	Item Name	Core: Baseline	Core: Add-On	Projected TCxP	Dependencies	Page
DE-1B	Various White Papers, Briefings, Prototypes	■		■		93
DE-2B	Guide to Best MBSE Practice: Edition 1	■		■	DE-12B	102
DE-3B	Guide to SW Architecture Assessment Practice	■		■		102
DE-4B	Dependable Real-Time System Technology (Simplex Architecture for Dependable Real-Time Software)	■		■	DE-7B	105
DE-5B	Open Systems Standards (for Real-Time Communication)	■		■		106
DE-6B	Report on Quality Attribute Tradeoffs	■				107
DE-7B	Airborne Radar Study Reports	■		■	DE-4B	107
DE-8B	Open Systems Handbook	■			DE-5B	107
DE-9B	Rate Monotonic Analysis (RMA) User's Forum	■				108
DE-10B	Roadmap for Environment Technology	■		■		114
DE-11B	Report on the State of the Practice in Process-Centered Environments	■		■		114
DE-12B	Electronic Software Engineering Information Base	■		■		114
DE-13B	Performance Engineering (EMM)	■			DE-6B, SP-1A, SP-2A, SP-3A	119
DE-14B	Reengineering Practice Guide	■		■	DE-2B, DE-12B	119
DE-1A	Evaluation of Architecture Representation and Analysis Technology		■	■	DE-3B, DE-6B, DE-4A	120
DE-2A	Case Studies of Software Architectures		■	■	DE-3B, DE-6B, DE-1A	121
DE-3A	Evaluation of a Commercial Application Development Environment Via Prototyping		■	■	DE-10B	121
DE-4A	System Composition Based on Software Architecture Principles		■			122
DE-5A	Community Work for Software Architecture Paradigm Shift		■	■	DE-1B, DE-12B	122
DE-6A	Dependable Real-Time Software System Demonstration		■	■	DE-4B, DE-7B	123
DE-7A	Dependable Real-Time Software System Handbook		■	■	DE-4B, DE-6A	123
DE-8A	Software Engineering Environments Technology Evaluation, Integration, and Measurement Initiative		■	■	DE-10B, DE-11B	124
DE-9A	Assessment of Collaboration Technology		■		DE-8B, DE-11B	125
DE-10A	Analysis and Transition of Application Generator Technology		■		DE-3A	126
DE-11A	X-Mosaic and the World Wide Web as an Enabler for Innovation Demonstration		■	■	DE-12B	126
DE-12A	Effectiveness of Software Engineering Information Base Technologies		■		DE-12B, DE-11A	127
DE-13A	Business Strategies for Model-Based Software Engineering (MBSE)		■	■	DE-2B, DE-3B, DE-3A	128
DE-14A	Prototype Design Record for a Multi-Supplier Software Component Industry		■		DE-12B, DE-14B, DE-12A	128

Figure A-3: Baseline and Add-On Items for the Disciplined Engineering Focus Area

Item Designator	Item Name	Core: Baseline	Core: Add-on	Projected TC&P	Dependencies	Page
TN-1A	Improving Software Design by Adding Security Engineering Principles		■			140
TN-2A	Security Improvement Plan for Software Engineering Environments		■			141

Figure A-4: Add-On Items for the Trustworthy Networks Focus Area

Item Designator	Item Name	Core: Baseline	Core: Add-on	Projected TC&P	Dependencies	Page
TT-1B	Methods and Practice of Software Technology Transition	■				152
TT-1A	Guide to Software Technology Transition Training and Continuing Education		■			153
TT-2A	Report on Best Practices in Software Technology Transition		■			154
TT-3A	Technology Transition Project Management Tool		■			154
TT-4A	Transition Assessment Instruments		■			155

Figure A-5: Baseline and Add-On Items for the Technology Transition Area

Item Designator	Item Name	Core		Project TC3P	Dependencies	Page
		Baseline	Add-on			
E-1B	Educational Products Advisory Board	■				159
E-2B	ETRB Operation	■				161
E-3B	Educational Process	■			E-2B	161
E-4B	Education 10th Year Retrospective	■				161
E-5B	Use NTU as Delivery Channel for Academic, Practitioner, and Leadership Courses	■			E-6B, E-7B	166
E-6B	MSE Core and Elective Courses	■				167
E-7B	Professional Education and Leadership Series	■				167
E-8B	8th CSEE	■				168
E-9B	IEEE/ACM Task Force on Software Engineering Profession	■				174
E-10B	Academic Influence	■				174
E-1A	Software Systems Engineering Course		■		E-2B, E-5B	175
E-2A	Real-Time Design Course		■		E-2B, E-5B	176
E-3A	Open Systems Course		■		E-2B	177
E-4A	Expansion of Leadership Series to NTU		■		E-2B, E-5B	178
E-5A	CD-ROM Packaging of Practitioner Series		■		E-2B	179
E-6A	Support for Level 3 Key Process Area (KPA) on Training		■			180
E-7A	Leadership on IEEE/ACM Task Force On Software Engineering Profession		■		E-9B	181
E-8A	Organizational Certification of Software Engineers (feasibility study)		■		E-9B	181

Figure A-6: Baseline and Add-On Items for the Education Area

Item Designator	Item Name	Core		Projected TC&P	Dependencies	Page
		Baseline	Add-on			
S-1B	Software Engineering Improvement Framework	■		■		192
S-2B	Draft Training Material for Software Engineering Improvement	■		■	S-1B	193

Figure A-7: Baseline Items for the Services Area

Item Designator	Item Name	Core		Projected TC&P	Dependencies	Page
		Baseline	Add-on			
CI-1B	Resident Affiliate Program	■				195
CI-2B	SPIN Coordination	■				196
CI-3B	SEPG National Meeting	■				200
CI-4B	SEI 1996 1&5 Year Plan	■			Core CDRL Requirement	—
CI-5B	Quarterly Update/Summary of Technical Operations	■			Core CDRL Requirement	—
CI-1A	Customer Relations		■			200

Figure A-8: Baseline and Add-On Items for the Customer Involvement Area

Appendix B Advisory Groups

The following advisory groups presently exist:

The Software Process Advisory Board oversees process products, services, and their supporting projects. It provides on-going advice concerning current and future strategic directions of the process focus area. Board meetings are held two times a year, with interim contact on specific subjects of relevance to the board's charter. Board members were carefully chosen for their expertise and experience; they are:

- 2 members from the Department of Defense (DoD)
- 2 members from the DoD contractor community
- 2 members from academia
- 1 member from Carnegie Mellon University (CMU)
- 3 former Software Engineering Institute (SEI) Process directors

The Capability Maturity Modeling (CMM) Advisory Board reviews proposed enhancements to the CMM and related products prior to their release, provides written recommendations to further the SEI mission and user satisfaction for the CMM and CMM-based products, provides advice on CMM product objectives and development plans, and facilitates acceptance of CMM products by the user community. The 11-member board consists of the CMM project leader, the Process Program director, and at least three members from both government and industry organizations.

The CMM-Based Appraisal (CBA) Advisory Board has recently been formed. It is made up of the previous Software Capability Evaluation (SCE) Advisory Board plus expanded membership to reflect the Software Process Assessment (SPA) community. The role of the board is to act as volunteer consultants on appraisal methods, their application, and their related products. The board consists of at least four government members, four industry members, and commercial appraisal vendors.

The Software Process Definition (SPD) Advisory Group includes approximately forty members from industry, government, and academia who are leading researchers and practitioners in software process. The group provides a forum for defining and debating process definition-related issues, refines objectives, evolves requirements, and reviews products.

The Software Process Measurement (SPM) Steering Committee provides technical input to the process measurement activities of the SEI. The steering committee is composed of 20 leaders in software measurement and management from industry, academia, government, and the SEI. The steering committee identifies and reviews measurement needs, activities and goals, progress, products, and future directions. The steering committee also promotes public acceptance of published results and work products.

The Software Capability Evaluation (SCE) Review Group, which represents the software community, began operating in early 1993 and provides a "selectively broad" review of the SCE method and its documents. The 32-member group includes the CBA Advisory Board, plus members from small businesses, academia, and other groups in the SEI working on related products. The SPA product set is presently receiving guidance from the **Software Process Assessment (SPA) Vendors Association**, a body represented by 10 industry groups, and from other SPA stakeholders from industry. Materials relating to appraisals, including questionnaires, are reviewed by the CBA Advisory Board, the SPA Vendors Association, and the CMM Advisory Board. As the appraisal methods continue to evolve with the CMM and are more integrated through the Common Rating Framework, these groups will continue to provide key guidance and advice.

The Systems Engineering CMM (SECMM) Steering Group oversees the Systems Engineering CMM. This group consists of six industrial participants, the SEI, ex-officio members from the National Council on Systems Engineering, and representatives from the U.S. government. The group meets four times per year, with interim contacts as needed. It has release authority over project work products and provides strategic direction on maintenance and expansion of the SECMM.

The People Management Capability Maturity Model (PMCMM) Advisory Board consists of two government and eight industry representatives. The members participate with the SEI in development and review of the PMCMM as well as provide oversight and guidance on other product development and direction.

The Educational Products Advisory Board provides advice to the SEI on its activities in support of software engineering education. The board comprises two representatives each from government, industry, and academia, and meets twice a year.

The Software Dependability Working Group (SDWG) is an informal spinoff from the Dependability Working Group, an activity started by the Aerospace Corporation. The SDWG is concerned with transitioning dependable software technology from the laboratory into practice. Members of the SDWG come from academia, government, and industry. To date the major "product" of the SDWG is the annual Dependable Software Technology Exchange, which has been held at the SEI for the past two years.

The CERT Advisory Task Force provides advocacy and community feedback on strategy and outputs to the trustworthy networks focus area. Task force members represent industry, government, and academia. Current members are from MCI and Unilever, the FBI, and Purdue University.

The following advisory groups will be meeting in the near future:

The Risk Management Advisory Board will be composed of respected members of the software engineering community who are knowledgeable about risk management. The members will be drawn from industry, government, and academia, as follows:

- Industry: Drawn from different corporations within the defense and aerospace sector or from different corporations within the commercial sector.
- Government: Drawn from different branches of the armed services and different agencies.
- Academia: Drawn from different universities. Having one member from CMU will be encouraged.

The Disciplined Engineering Advisory Board will provide advocacy and community feedback on strategy and outputs. The board will comprise respected members of the software engineering community who are knowledgeable about disciplined engineering. The members will be drawn from industry, government, and academia, as follows:

- Industry: Drawn from different corporations within the defense and aerospace sector or from different corporations within the commercial sector.
- Government: Drawn from different branches of the armed services and different agencies.
- Academia: Drawn from different universities. Having one member from CMU will be encouraged.

Appendix C Resident Affiliates

	Name	Total to Date	One Affiliate in Residence as of 1 July 1994
INDUSTRY	AT&T Bell Labs	1	
	Boeing	1	
	Computer Sciences Corporation	5	▮
	GE Aerospace	2	
	General Dynamics	1	
	GTE Government Systems	3	▮
	Hughes Aircraft Company	5	▮
	Lockheed Missiles & Space Co., Inc.	1	
	Loral Federal Systems	5	
	Pacific Bell	1	
	Process, Inc.	1	
	Raytheon Company	1	
	SEMATECH	1	▮
	Siemens Corporate Research	1	▮
	SYSCON Corporation	1	
	TeleSoft	1	
	Texas Instruments	2	▮
	Unisys	3	▮
	Westinghouse Electric Corporation	3	
	Wilcox Electric	1	▮

		Name	Total to Date	One Affiliate in Residence as of 1 July 1994
GOVERNMENT	Navy	Coastal Systems Station	1	
		Naval Air Development Center	2	
		Naval Ocean Systems Center	3	
		Naval Surface Warfare Center	3	
		Naval Undersea Warfare Engineering Station	1	
		Naval Undersea Weapons Engineering Station	1	
		Naval Weapon Center	2	
		Naval Air Warfare Center	1	
	Army	Communications-Electronics Command	6	
		United States Military Academy	1	■
	Air Force	Air Combat Command	1	
		Air Force Institute of Technology	5	
		Air Logistics Center	1	
		Electronic Systems Center	3	
		Space Command	1	
		Standard Systems Center	1	
	Other	Department of Defense	9	■

Appendix D Technical Collaboration

D.1 Current Cooperative Research and Development Agreements (CRADA)[1]

Revenue

(1)	Union Switch and Signal
(1)	University of Houston Clear Lake

Commercialization (Royalties)

(1)	Abacus/Institute for Software Process Improvement
(1)	Integrated Systems Diagnostics, Inc.
(1)	Process Focus Management

In-Kind

(1)	LoneWolf Systems, Inc.

D.2 Current Technical Collaboration Partners[1]

(1)	Allied Signal Aerospace	(4)	Loral Federal Systems
(1)	Applied Software Engineering Centre, Canada	(1)	Master Systems
(1)	Bell Northern Research	(1)	Motorola
(1)	Boeing	(1)	National Security Agency
(1)	Center for Naval Analysis	(1)	Northrop

1. Current as of 6/29/94

(2)	Citibank	(1)	Process Enhancement Partners Inc.
(1)	Computer Sciences Corporation	(1)	Schlumberger
(1)	Computing Devices International, Inc.	(1)	Science Applications International Corporation
(1)	Electronic Data Systems	(1)	SEMATECH
(1)	Federal Express	(2)	Siemens Corporate Research
(1)	Fisher Rosemont	(1)	Software Productivity Consortium
(1)	Ford	(6)	Texas Instruments
(1)	Grumman	(1)	US West
(1)	Harris Corporation	(2)	Unisys
(3)	Hewlett-Packard	(1)	Universidad Politecnica de Madrid, Spain
(4)	Hughes	(1)	UNUM Life Insurance Company of America
(1)	Institute for Software Process Improvement	(1)	University of Southern California Center for Software Engineering
(1)	Kodak	(1)	Warner Robbins AFB
(3)	Lockheed	(1)	Westinghouse
(1)	Logicon	(2)	Xerox

References

[Bell 71] C.G. Bell and A. Newell. *Computer Structures: Readings and Examples.* New York: McGraw-Hill, 1971.

[Conner 82] Conner, D. R.; and Patterson, R. W. "Building Commitment to Organizational Change." *Training and Development Journal,* April 1982.

[Davis 71] See Davis, H.R., "A Checklist for Change," in National Institute of Mental Health, *A Manual for Research Utilization*, Washington, DC: U.S. Government Printing Office, 1971.

[Davis 79] Davis, H.R. & Salasin, S.E., "*Change: Decisions and Their Implementation,*" in S.E. Feldman (Ed.), The Administration of Mental Health Services (Rev. ed.), Springfield, IL: Thomas, 1979.

[Dion 92] Dion, R. "Elements of a Process-Improvement Program (Software Quality)." *IEEE Software* 9, 4 (July 1992): 83-85.

[Fowler 93a] Fowler, P. & Levine, L., *Conceptual Framework for Software Technology Transition* (CMU/SEI-93-TR-31). Pittsburgh, PA.: Software Engineering Institute, Carnegie Mellon University, 1993.

[Fowler 93b] Fowler, P. & Levine, L. *Technology Transition Push: A Case Study of Rate Monotonic Analysis (Part 1)* (CMU/SEI-93-TR-29). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993.

[Grady 89] Grady, R. B., and Caswell, D. L. *Software Metrics: Establishing a Company-Wide Program.* Englewood Cliffs, N.J.: Prentice-Hall, 1989.

[Humphrey 91] Humphrey, W. S.; Snyder, T. R.; and Willis, R. R. "Software Process Improvement at Hughes Aircraft." *IEEE Software* 8, 4 (July 1991): 11-23.

[Leonard-Barton 88] Leonard-Barton, D., "Implementation as Mutual Adaptation of Technology and Organization." *Research Policy 17,* 5 (October 1988), 102-110.

[Lipke 92] Lipke, W. H.; and Butler, K. L. "Software Process Improvement: A Success Story." *CrossTalk*, November 1992.

[Moore 91] Moore, G. *Crossing the Chasm: Marketing and Selling Technology Products to Mainstream Customers.* Harper Business, 1991.

[Paul 89] Paul, J.H., & Simon, G. C. (Eds.). *Bugs in the Program: Problems in Federal Government Computer Software Development and Regulation.* U.S. House of Representatives, Committee on Science, Space, and Technology, 1989.

[Putnam 93] Putnam, L. H. *The Economic Value of Moving Up the SEI Scale.* Technical Report, Quantitative Software Management Inc., McLean, VA., 1993.

[Redwine 84] Redwine et al, *DoD Related Software Technology Requirements, Practices, and Prospects for the Future.* (Technical Report IDA Paper P-1788), Institute for Defense Analysis, 1984.

[Rogers 83] Rogers, E. *Diffusion of Innovation.* New York, Free Press, 1983.

[SEI 94] *The Blue Ribbon Comprehensive Review of the Software Engineering Institute*, Software Engineering Institute, Pittsburgh, PA.: January 1994.

[Tornatzky 90] Tornatzky, L.G. & Fleischer, M., *The Processes of Technological Innovation.* Lexington, MA, Lexington Books, 1990.

[Willis 83] Willis, R., "Technology Transfer Takes 6 +/- 2 Years," In R. Morton (Ed.), *Proceedings of the IEEE Computer Society Workshop on Software Engineering Technology Transfer* (pp.108-117), 1983.

[Wohlwend 93] Wohlwend, H.; and Rosenbaum, S. "Software Improvements in an International Company." *Proceedings of the 15th International Conference on Software Engineering.* Baltimore, MD, May 1993.

List of Acronyms

ACM	Association of Computing Machinery
AFMC	Air Force Materiel Command
AFSPACECOM	Air Force Space Command
AMC	Army Materiel Command
ARM	Acquisition Risk Management
ARPA	Advanced Research Projects Agency
BPG	Baseline Practices Guide
CARDS	Central Archive for Reusable Defense Software
CASE	computer-aided software engineering
CBA	CMM-based appraisal
CBA-IPI	CMM-based appraisal for internal process improvement
CBA-SCE	CMM-based appraisal for software capability evaluation
CERT	Computer Emergency Response Team
CMM	capability maturity model
CMU	Carnegie Mellon University
COTS	commercial off-the-shelf
CRF	common rating framework
CS	Computer Society
CSEE	Conference on Software Engineering Education
CSLB	California State at Long Beach
DISA	Defense Information Systems Agency
DMA	Defense Mapping Agency

DoD	Department of Defense
DSMC	Defense Systems Management College
EM	educational materials
EMM	engineering maturity model
ESIP	Embedded Computer Resources Support Improvement Program
ETRB	Education and Training Review Board
FIRST	Forum of Incident Response and Security Teams
GAO	Government Accounting Office
GUI	graphical user interface
IPI	internal process improvement
ISI	Information System Institute
ISO	International Standards Organization
JAC	Joint Advisory Committee
KPA	key process areas
LAS	Aircraft Software Division
MATA	Media and Arts Technology Alliance
MBSE	model-based software engineering
MCCR	mission-critical computer resources community
MCTSSA	Marine Corps Tactical Systems Support Agency
MSE	Master of Software Engineering
MTTD	mean time to defect
NAVOCEANO	Naval Oceanic Office
NAWC	Naval Air Warfare Center

NCOSE	National Council on Systems Engineering
NGCR	Next Generation Computer Resources
NII	National Information Infrastructure
NIST	National Institute of Standards and Technology
NLP	natural language processing
NOAA	National Oceanic and Atmospheric Administration
NSF	National Science Foundation
NTP	National Technology Policy
NTU	National Technological University
OC-ALC	Oklahoma Air Logistics Center
OSC	Office of the Secretary of Defense
P/TSP	personal/team software process
PAE	product attribute engineering
PDSS	post-deployment software support
PEO (MIS)	Air Force Program Executive Office for Management Information Systems
PEO	program executive officers
PM	program manager
PMCMM	people management capability maturity model
QIP	quality improvement process
R&D	research and development
RMA	rate monotonic analysis
ROI	return on investment
SACMM	systems acquisition capability maturity model

SCE	software capability evaluation
SCS	School of Computer Science
SECMM	systems engineering capability maturity model
SEE	software engineering environment
SEI	Software Engineering Institute
SEI	Software Engineering Institute
SEPG	software engineering process group
SET	software engineering techniques
SIGSCE	Special Interest Group on Computer Science Education
SPA	software process assessment
SPAWAR	Space and Naval Warfare Systems Command
SPI	software process improvement
SPICE	Software Process Improvement Capability dEtermination
SRE	software risk evaluation
SSC	Standard Systems Center
STARS	Software Technology for Adaptable, Reliable Systems
STSC	Software Technology Support Center
TCA	technical collaboration agreement
TO&P	technical objectives and plans

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release Distribution Unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CMU/SEI-94-SR-19	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Software Engineering Institute 6b. OFFICE SYMBOL (if applicable) SEI	7a. NAME OF MONITORING ORGANIZATION SEI Joint Program Office
6c. ADDRESS (city, state, and zip code) Carnegie Mellon University Pittsburgh PA 15213	7b. ADDRESS (city, state, and zip code) HQ ESC/ENS 5 Eglin Street Hanscom AFB, MA 01731-2116
8a. NAME OFFUNDING/SPONSORING ORGANIZATION SEI Joint Program Office 8b. OFFICE SYMBOL (if applicable) ESC/ENS	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F1962890C0003
8c. ADDRESS (city, state, and zip code)) Carnegie Mellon University Pittsburgh PA 15213	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO: 63756E PROJECT NO.: N/A TASK NO: N/A WORK UNIT NO.: N/A

11. TITLE (Include Security Classification)
Draft SEI Program Plans: 1995-1999

12. PERSONAL AUTHOR(S)

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (year, month, day)	15. PAGE COUNT
Final	FROM TO	August 1994	242

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (continue on reverse of necessary and identify by block number)
program plans, technical program, strategy, one-year plan, five-year plan, technology transition, products and services.

19. ABSTRACT (continue on reverse if necessary and identify by block number)
This document presents the Software Engineering Institute (SEI) strategy and one-year implementation plan for calendar year (CY) 1995, together with the SEI five-year program plan. This document is, in essence, a proposal. It describes the strategic directions and offers detailed options for the coming year. Until the proposed options are selected and budget allocations are approved by the sponsor, the SEI cannot commit to specific work or supporting schedules.

(please turn over)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ■ SAME AS RPT ☐ DTIC USERS ■	21. ABSTRACT SECURITY CLASSIFICATION Unclassified, Unlimited Distribution
22a. NAME OF RESPONSIBLE INDIVIDUAL Thomas R. Miller, Lt Col, USAF	22b. TELEPHONE NUMBER (include area code) (412) 268-7631 22c. OFFICE SYMBOL ESC/ENS (SEI)

ABSTRACT — continued from page one, block 19